OMNICAL

SECOND SEMESTER FINAL DOCUMENTATION

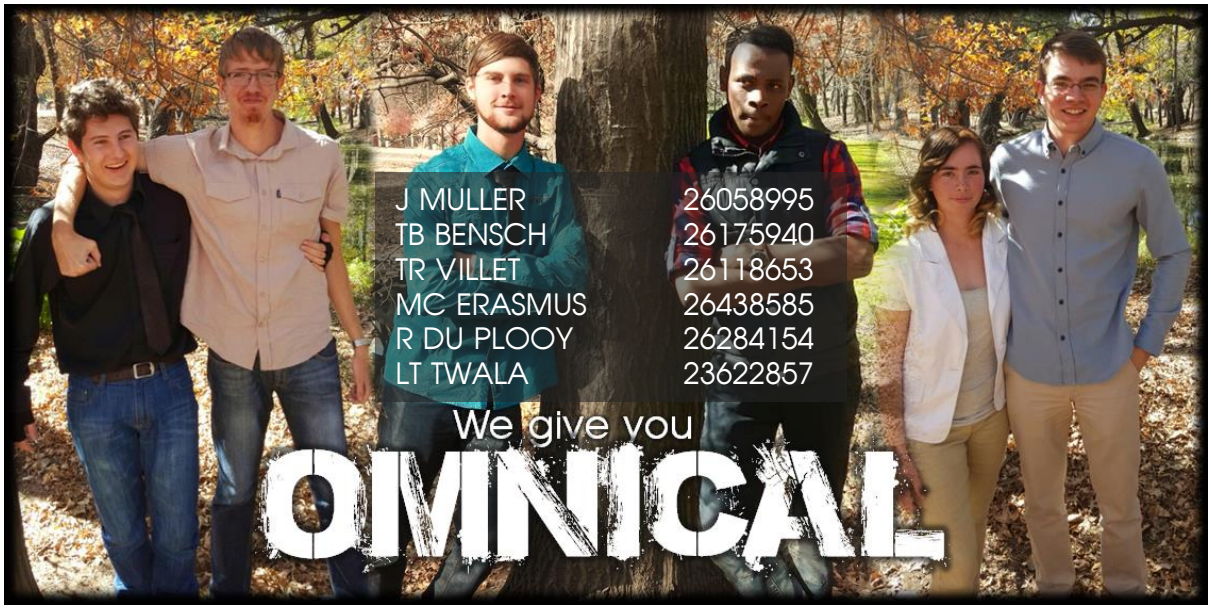


Table of Contents

OBJECT ORIENTATED ANALYSIS

Refined Use Case list:

Event	**Response**
Receive information from database	Database sends information to where it is needed
Send information to database	Database receives information from where it was created, changed or used
Create event	Data is created to be sent to the database.
View timetable	Data is accessed and can be viewed

Compiled By: J Muller

1. Potentials Objects:

frmAnnouncements
frmCalendar
frmCalMyFriend
frmEditTimetable
frmHome
frmLicenseAgreement
frmLogin
frmRegister
frmPreferences
frmSettings
frmViewData
frmViewStatistics
frmViewTimetable
frmUpdateAdmin
frmUpdateProfile

Compiled By: J Muller

2. Select proposed objects by cleaning the potential objects

Question comparisons

Proposed objects	Synonym?	Outside scope?	External role?	Unclear?	Action?
frmAnnouncements	✓	✓	✓	×	✓
frmCalendar	✓	✓	✓	×	✓
frmCalMyFriend	×	✓	✓	×	✓
frmEditModule	✓	×	✓	×	✓
frmEditTimetable	✓	✓	✓	×	✓
frmHome	✓	✓	✓	×	✓
frmLicenseAgreement	✓	✓	×	×	✓
frmLogin	✓	×	✓	×	✓
frmRegister	✓	×	✓	×	✓
frmPreferences	×	✓	✓	×	✓
frmSettings	×	✓	✓	×	✓
frmViewData	✓	×	✓	×	✓
frmViewStatistics	×	✓	✓	×	✓
frmViewTimetable	✓	×	✓	×	✓

frmUpdateAdmin	✓	✓	✓	×	✓
frmUpdateProfile	✓	✓	✓	×	✓

Object List:

Proposed objects	Result	Reason
frmAnnouncements	×	Outside of scope
frmCalendar	×	Outside of scope
frmCalMyFriend	×	Outside of scope
frmEditModule	✓	Renamed "Module"
frmEditTimetable	✓	Renamed "Schedule
frmHome	×	Synonym of "frmLogin" divided into Admin, Student and Staff
frmLicenseAgreement	×	Outside of scope
frmLogin	✓	Renamed as User
frmRegister	×	Outside of scope
frmPreferences	×	Outside of scope
frmSettings	✓	Renamed "Settings"
frmViewData	×	Outside of scope
frmViewStatistics	×	Outside of scope
frmViewTimetable	✓	Renamed "Timetable"
frmUpdateAdmin	×	Synonym of "Login" renamed User
frmUpdateProfile	×	Synonym of "Login" renamed User

Compiled By: J Muller

Proposed Object List:

Admin
Module
Roster
Schedule
Settings
Staff
Student
Timetable
User

Compiled By: J Muller

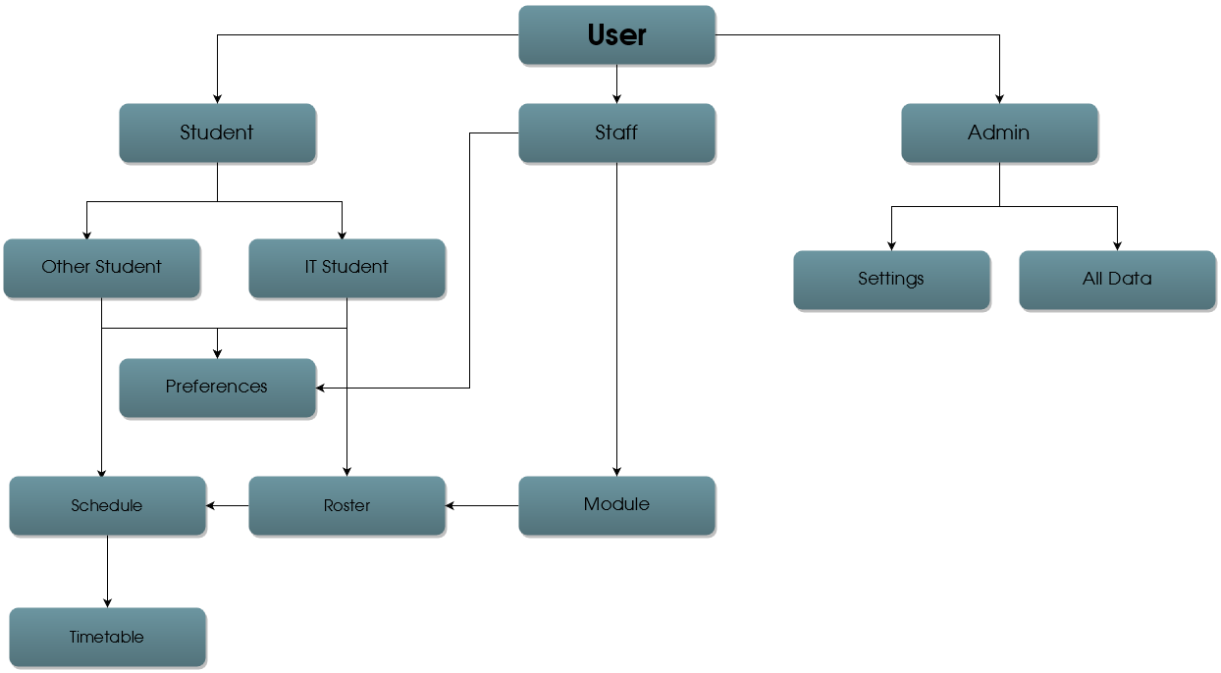
3. Create associations between objects

Association and Multiplicity

	User	Timetable	Settings
User		Student creates zero to many timetables	Admin creates one or many passwords
Timetable	Is created by one and only one student		XXX
Settings	Is created by one and only one admin	XXX	

Compiled By: J Muller

Generalisation and Specification



Compiled By: J Muller

Aggregation/ Composition Relationships

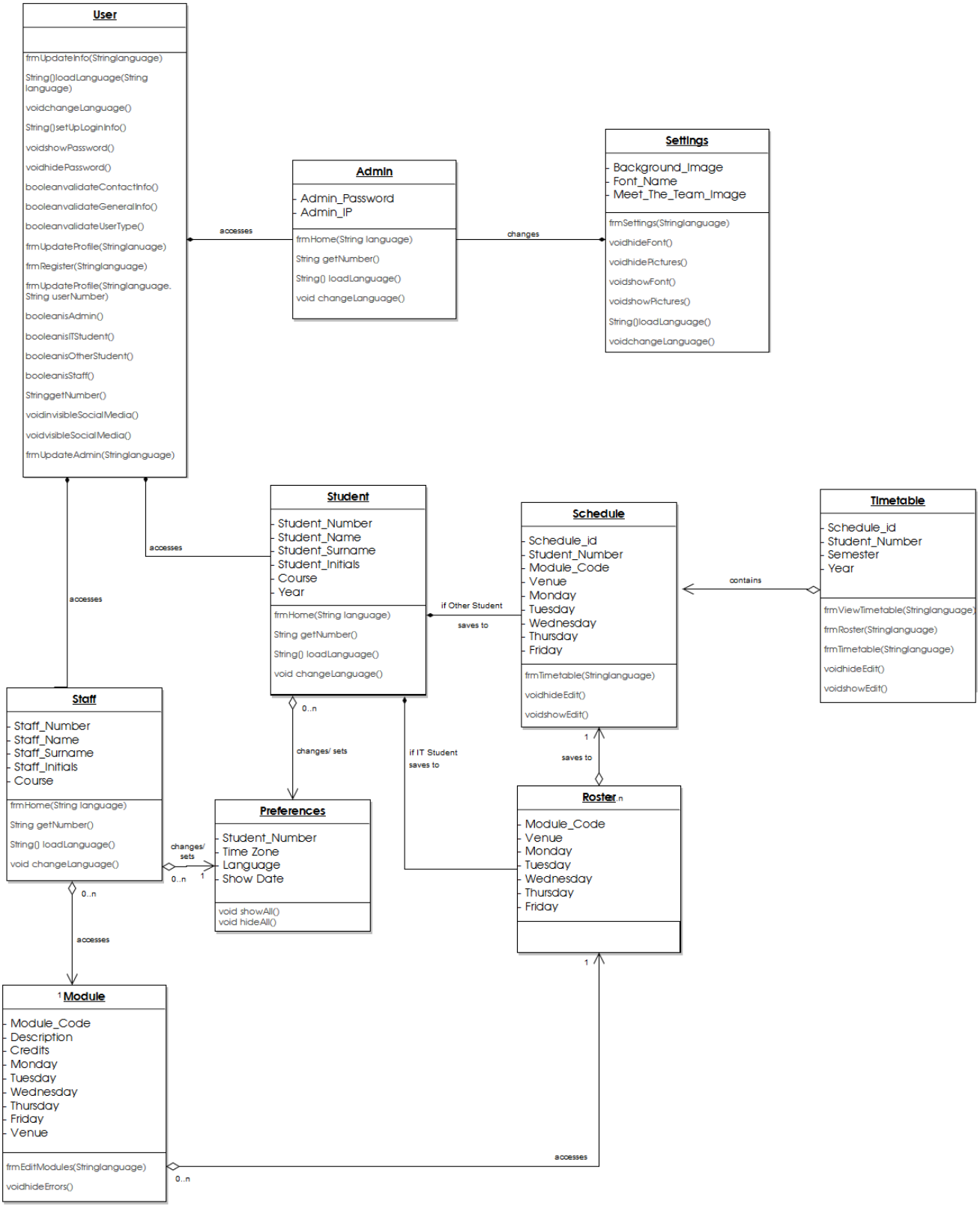
- Timetable contains Schedule and is therefore, *an aggregate relationship.*
    - Due to the fact that Schedule is part of Timetable but Timetable is not part of Schedule, this represents a whole/part aggregate relationship.
- The relationship between Preferences and IT Students/ Other Students / staff is an *aggregate relationship.*
    - Preferences form part of both IT Student, Other Student and Staff but It Student, Other Student and Staff do not form part of Preferences
- The relationship between Roster and Schedule is *aggregate.*
    - If Roster dies, Schedule won't die, but it Schedule dies, Roster dies.
- Likewise, the relationship between Staff and Module is *aggregate.*
    - If Module dies, Staff won't die, but it Staff dies, Module dies.
- The relationship between Module and Roster is an *aggregate relationship.*
    - If Roster dies, Module won't die, but it Module dies, Roster dies.
- User contains Student, Staff and Admin. These subclasses each contain their own objects. This represents a *composition relationship*
    - If User were to be destroyed, all other paths would be destroyed with it.
- Other student and student also represent a *composition relationship*
    - If Student were to be destroyed so would the attached strings
- The relationship between Other Student and Schedule is a *composition relationship.*
    - If Other Student dies, so do the paths underneath it, this affects the Schedule path.
- Likewise, IT Student and Roster also represent a *composition relationship*
    - If IT Student dies so does the path to Roster.
- The relationship between Schedule and Timetable are *composite.*
    - If Schedule dies so does the path to Timetable.
- The relationships between Admin and Settings/ All data are *composite.*
    - If Admin dies so do the paths to Settings and All Data

4. Relevant Methods:

- User
    - frmUpdateInfo(String language)
    - String() loadLanguage(String language)
    - void changeLanguage()
    - String() setUpLoginInfo()
    - void showPassword()
    - void hidePassword()
    - boolean validateContactInfo()
    - boolean validateGeneralInfo()
    - boolean validateUserType()
    - frmUpdateProfile(String lanuage)
    - frmRegister(String language)
    - frmUpdateProfile(String language, String userNumber)
    - boolean isAdmin()
    - boolean isITStudent()
    - boolean isOtherStudent()
    - boolean isStaff()
    - String getNumber()
    - void invisibleSocialMedia()
    - void visibleSocialMedia()
    - frmUpdateAdmin(String language)
- Admin, Staff, Student
    - frmHome(String language)
    - String getNumber()
    - String() loadLanguage()
    - void changeLanguage()
- Module
    - frmEditModules(String language)
    - void hideErrors()
- Schedule
    - frmTimetable(String language)
    - void hideEdit()
    - void showEdit()
- Timetable
    - frmViewTimetable(String language)
    - frmRoster(String language)
    - frmTimetable(String language)
    - void hideEdit()
    - void showEdit()
- Settings
    - frmSettings(String language)
    - void hideFont()
    - void hidePictures()
    - void showFont()
    - void showPictures()
    - String() loadLanguage()
    - void changeLanguage()

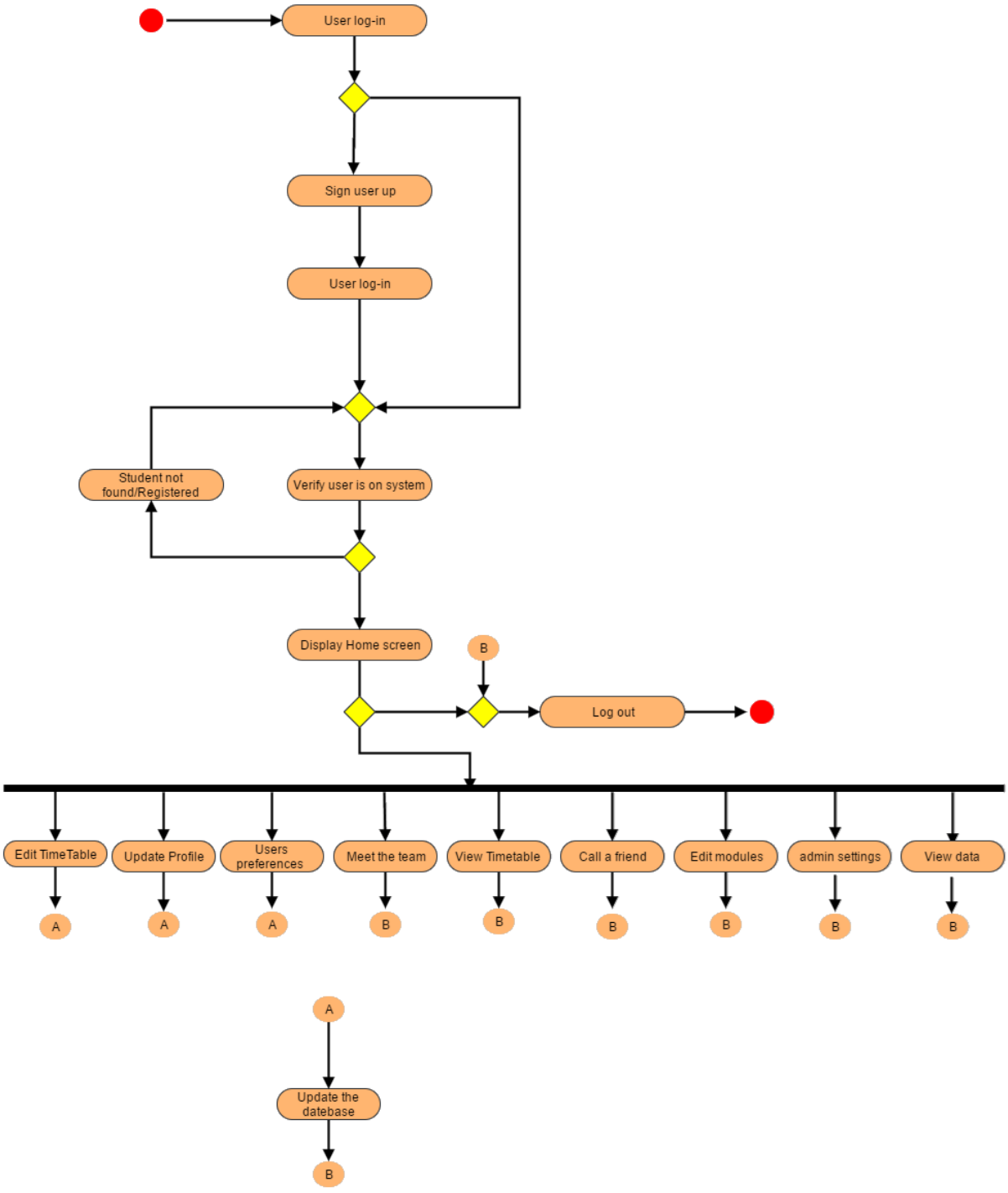
Compiled By: J Muller

Class Diagram



By: J Muller

Activity Diagram



By: R Du Plooy

SYSTEM DESIGN

System Analysis vs System Design

System Analysis Aspects	Mapping and description of mapping	System Design Aspects	System Analysis aspects to be improved
1. Scope Definition	1a When conducting the physical design and integration phase of system design, the following must be taken into account regarding scope definition: 1. Identification of baseline problems and opportunities 2. Negotiation of baseline scope 3. Assessment of baseline project worthiness 4. Development of baseline schedule and budget 5. Communication of the project plan. Regarding procurement of software and services, the following must be taken into account: 1. Research technical criteria and options 2. Solicit proposals or quotes from vendors	a. Procurement (of software and services)	N/A
2. Problem Analysis	2c When conducting the physical design and integration phase of system design, the following must be taken into account regarding problem analysis: 1. The understanding of the problem domain 2. The analysis of problems and opportunities 3. The analysis of business processes 4. The establishment of system improvement objectives 5. The updated or redefined project plan 6. The communication of findings and recommendations	b. Decision Analysis (for integration)	In terms of system development, the more we improve our logical design, the more we solve problems which we didn't notice during this system analysis phase (for example, changing administration settings for security purposes)
3. Requirement Analysis	3b When conducting the physical design and integration phase of system design, the following	c. Decision Analysis (for software and services)	Due to the improvement in problem analysis, requirements need

	must be taken into account regarding requirement analysis: 1. The identification and expression of system requirements 2. The prioritization of system requirements 3. The updated or redefined project plan 4. The communication of the requirements statement Regarding decision analysis for integration, the following must be taken into account: 1. The validation of vendor claims and performances 2. The evaluation and ranking vendor proposals 3. The awarding of contracts and debriefing vendors		to be adjusted accordingly (for example, system requirements need to be adjusted to specific administration requirements like credentials)
4. Logical Design	4e When conducting the physical design and integration phase of system design, the following must be taken into account regarding logical design: 1a. The structure of functional requirements 1b. The prototyping of functional requirements 2. The validation of functional requirements 3. The definition of the acceptance test cases	d. Implementation of software)	Logical design is improved in order to bring all analysis together and simultaneously provide the user with a fully functional system. As there are some inconsistencies between a few system analysis phases, the phases need to be improved so that we have consistency when the logical design is completed (for example, recording and allowing the change of administration credentials)
5. Decision Analysis	5d When conducting the physical design and integration phase of system design, the following must be taken into account regarding decision analysis: 1. The identification of candidate solutions 2. The analysis of candidate solutions 3. The comparison of candidate solutions	e. Design (and integration)	N/A

	4. The updated or redefined project plan 5. The recommendation of a system solution The decision phase of system design takes the following into consideration: 1. The design of the application architecture 2. The design of the system databases 3. The design of the system interface 4. The packaging design specifications 5. The updated project plan		

By: J Muller

12 Point Plan for Improvement

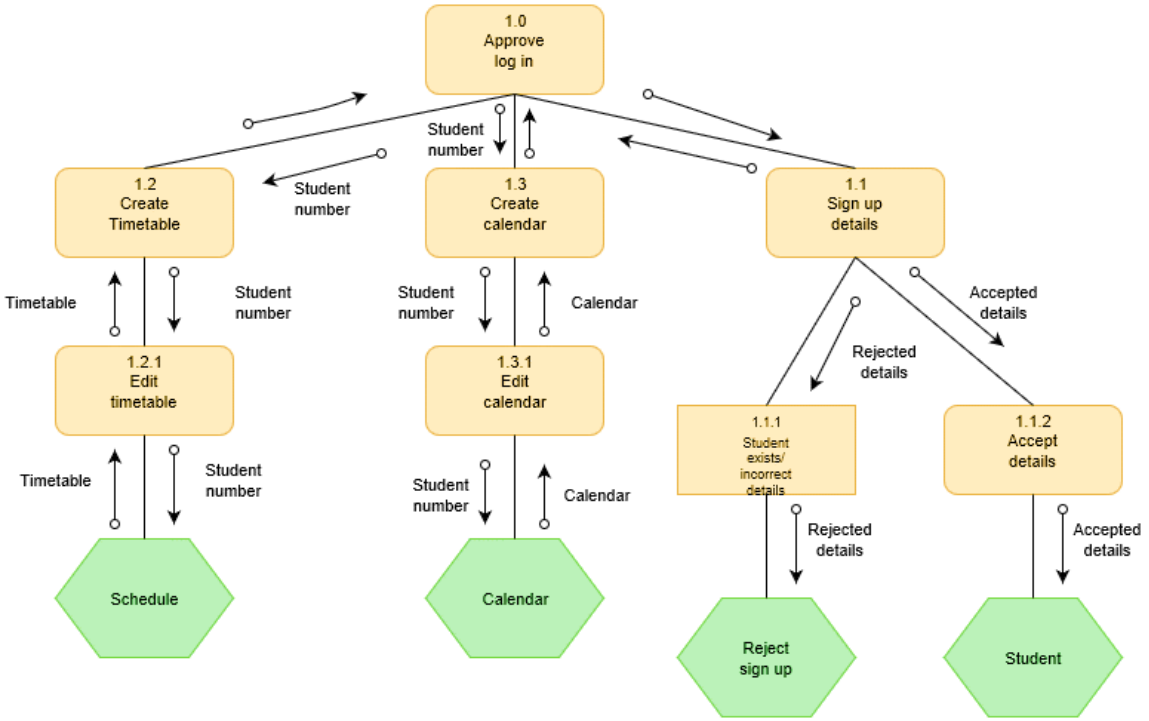
Before system design took place, the first 5 phases of the system analysis process was mapped against the tasks of the procurement phase representing our system design.

After analysing the system analysis aspects, we created a 12 point plan to better the aspects that needed improvement:

1. Add a registration process
2. Add a home screen
3. Add a "View Timetable Out Of Editor" option
4. Add a "View Statistics" option
5. Add a "User Preferences" option
6. Add an "Edit Modules" option for staff members and limit their access
7. Add a "Meet The Team" option
8. Add a "View in Calendar" option
9. Add a "Update profile" option
10. Add a "Settings" option for admin only
11. Add a "View data" option for admin only
12. Add an "Update Admin Info" option

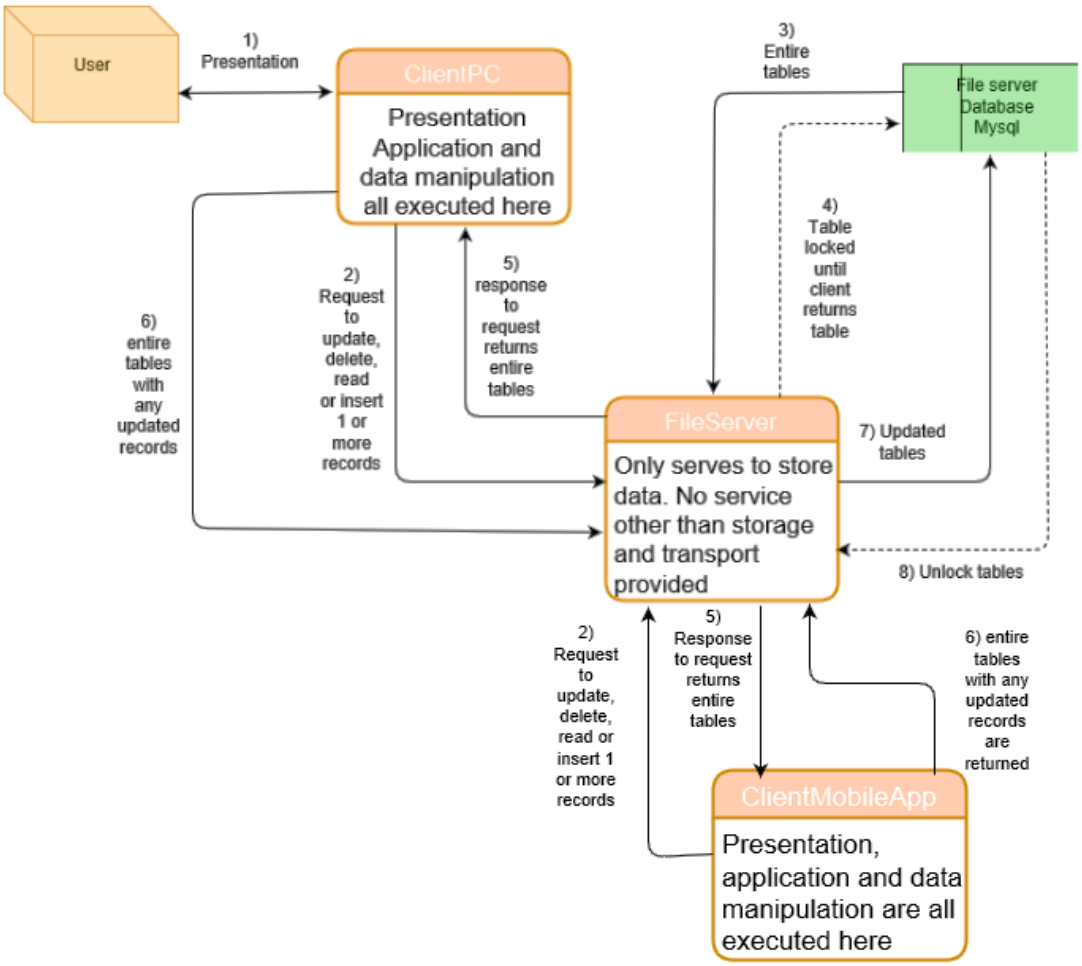
By: J Muller

Structured Diagram



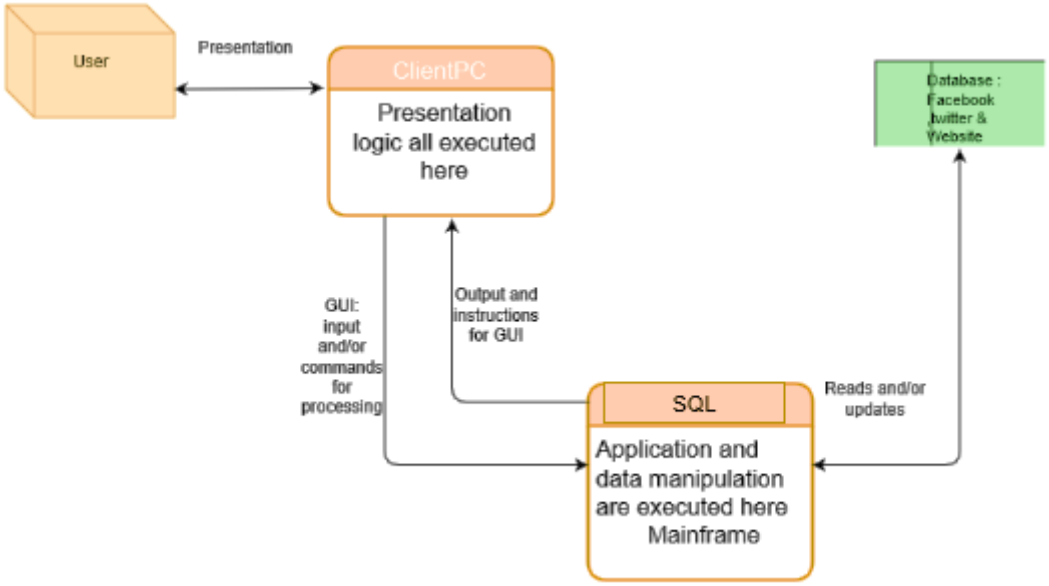
By: TB Bensch

File/Server Architecture



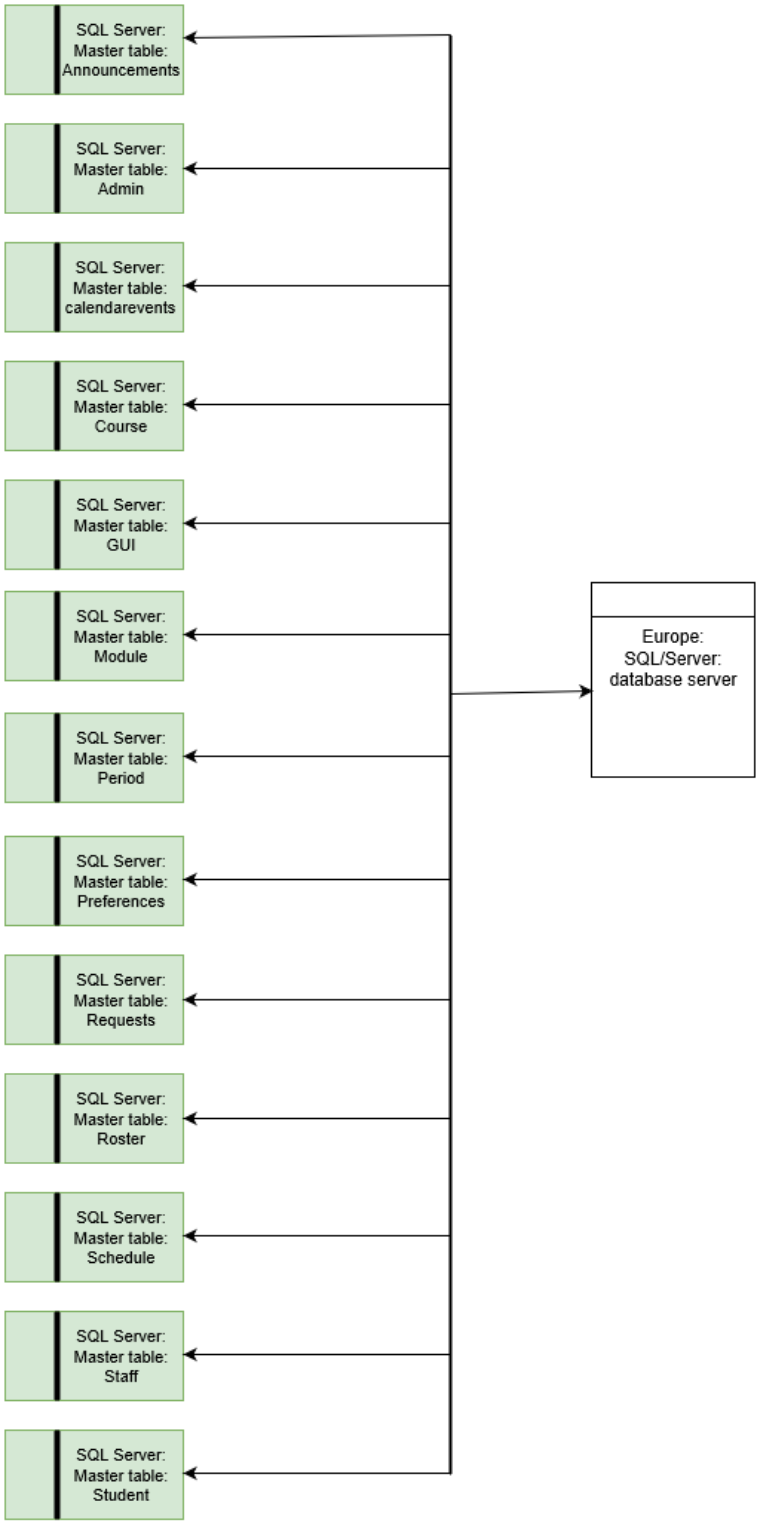
By: TB Bensch

Client/Server Architecture



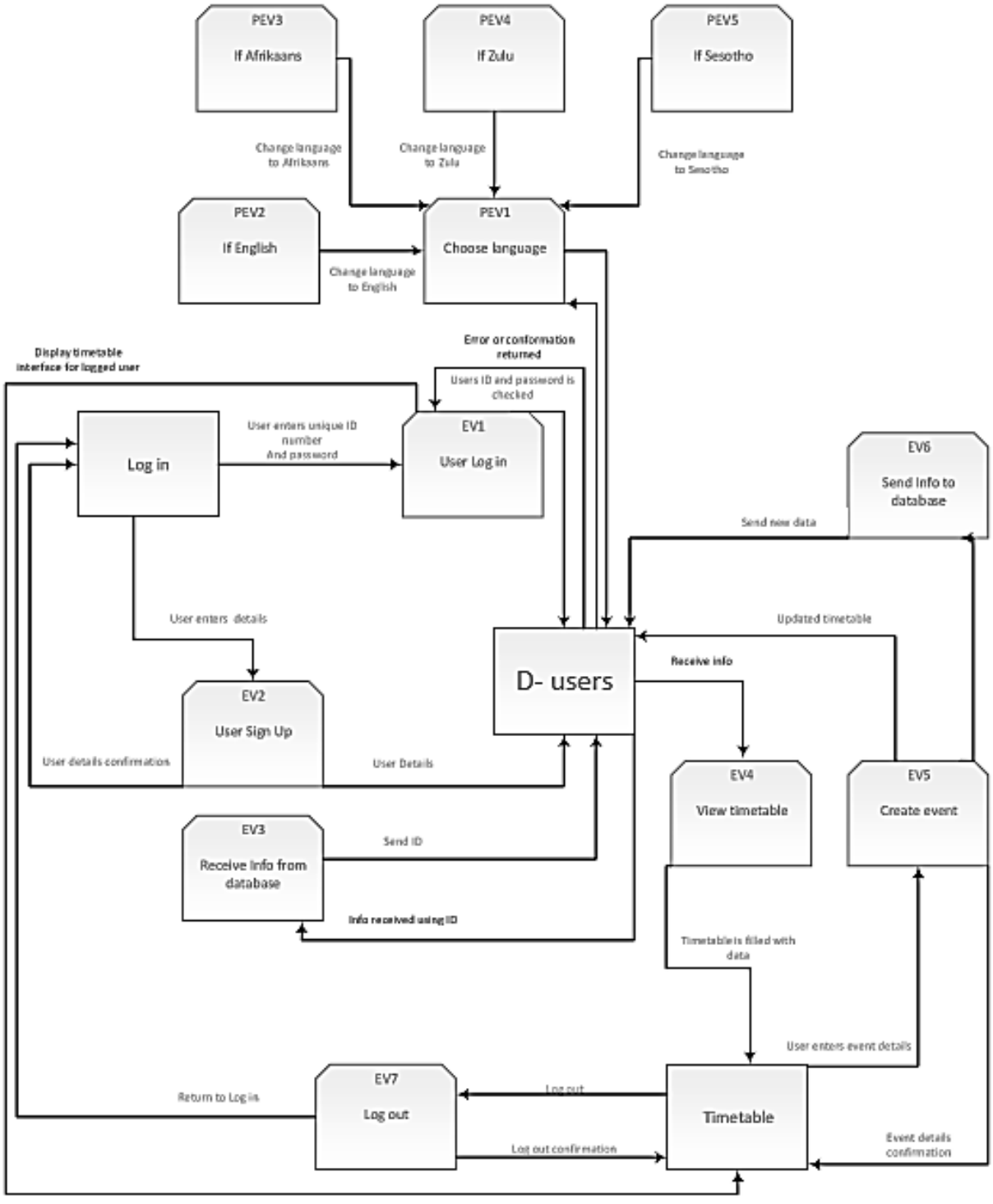
By: TB Bensch

Data Distribution Diagram



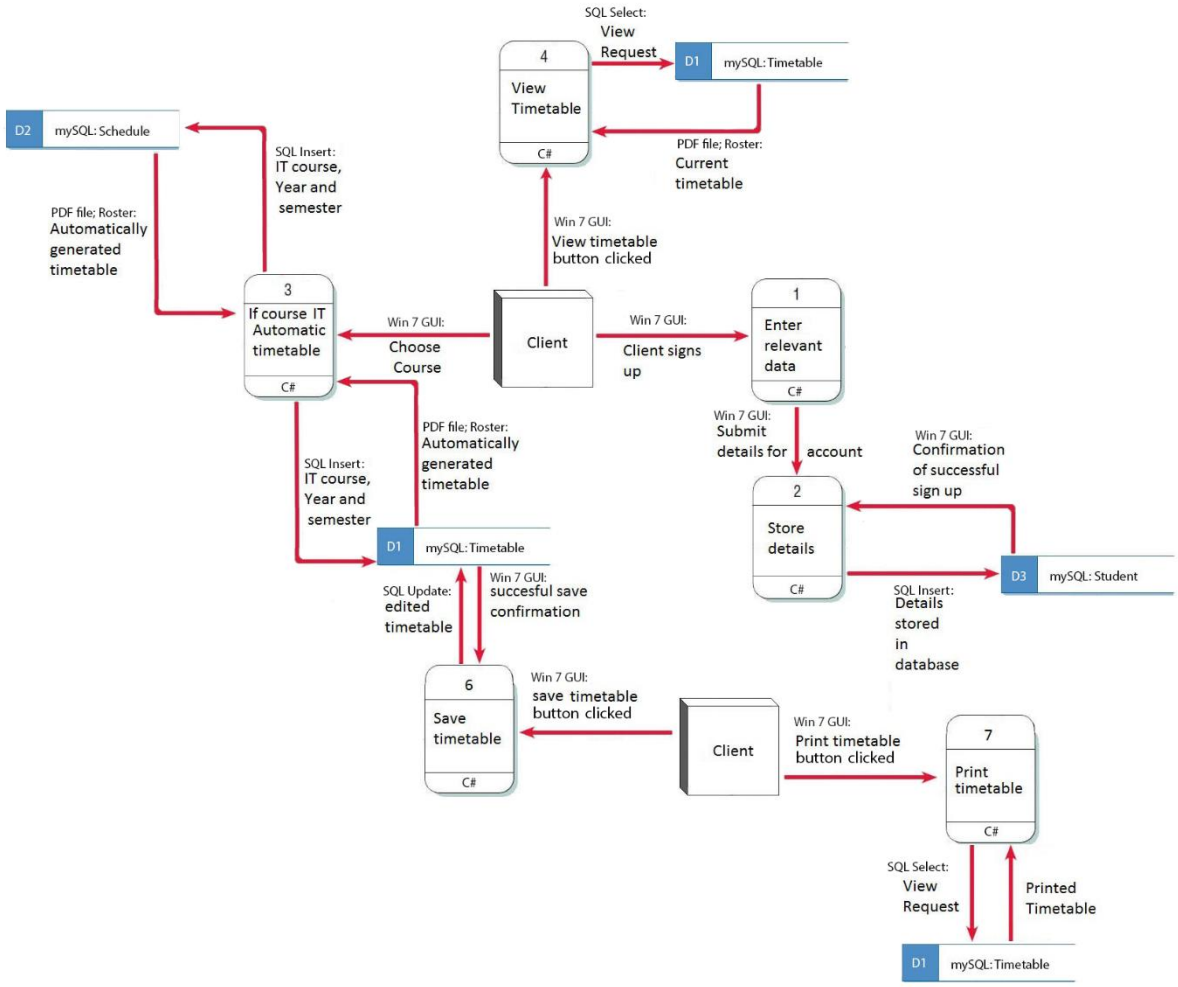
By: TB Bensch

Logical Data Flow Diagram



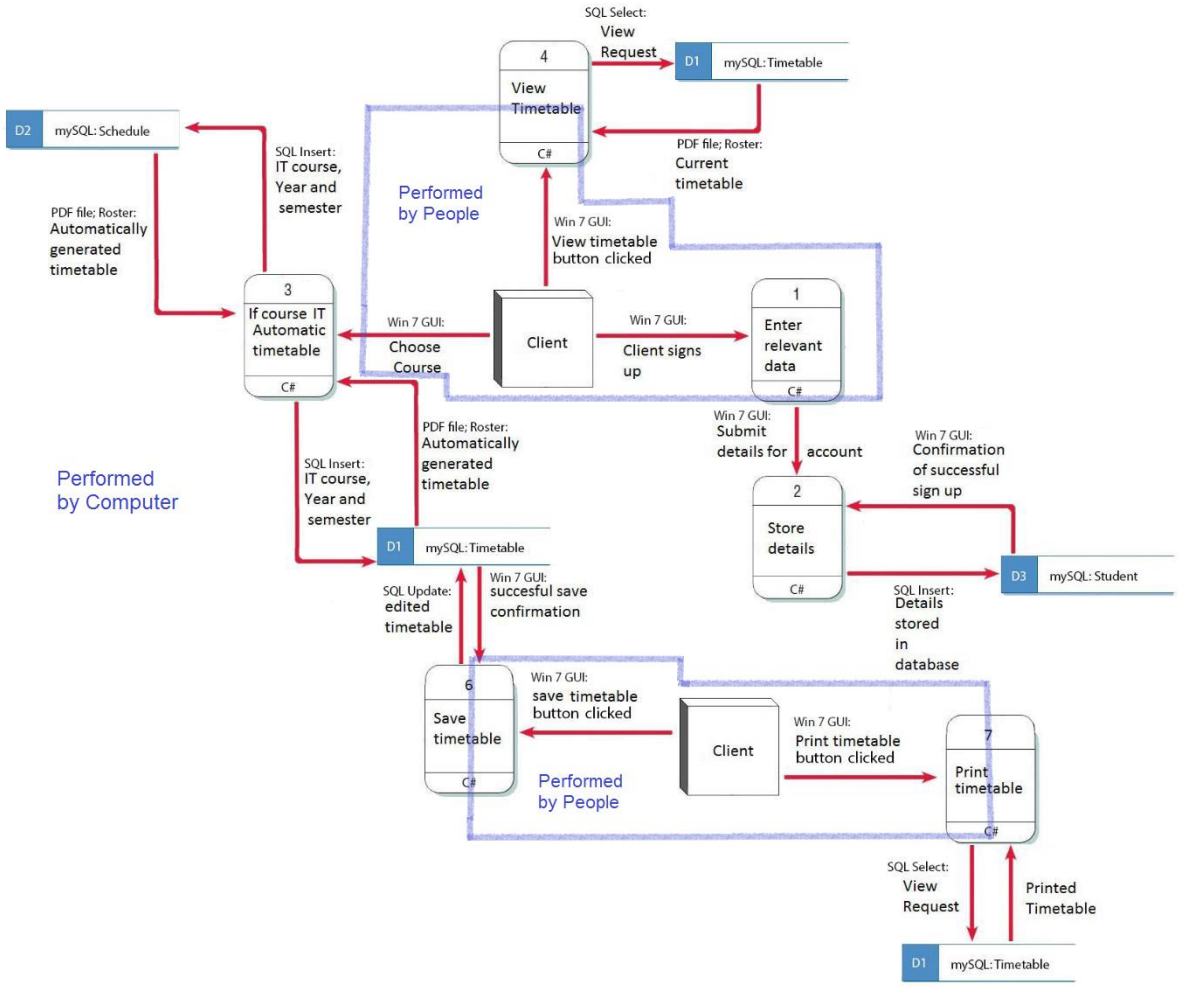
By: J Muller & MC Erasmus

Physical Data Flow Diagram



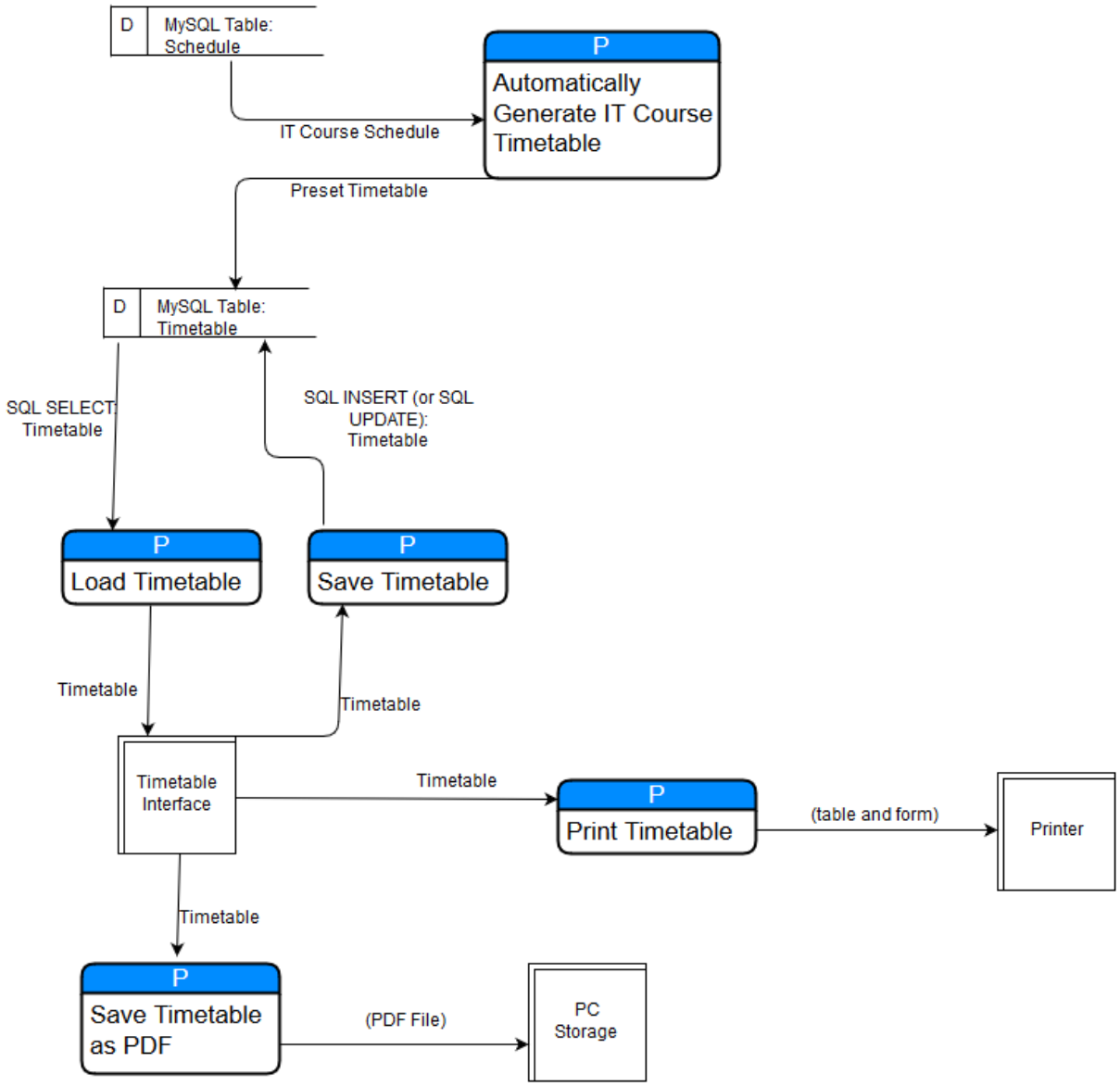
By: MC Erasmus and TR Villet

Person/Machine Boundary



By: MC Erasmus

Manual Design Unit



By: MC Erasmus

Application Architecture

What is application architecture? :

Application Architecture is a design that provides guidelines to application design. The design and patterns are influenced by principles which are provided by the application architectural paradigms. The name software architecture is frequently used to describe the design in terms of Information Technology.

The architectural design illustrates the patterns and distributions across the internet. The design comprises of principles in terms of high level rules that will control the functionality and capabilities of the system as set by the architecture. The distribution of data within the design has to be indicated on how the flow or pattern will be distributed within the databases.

An application architecture will have to comprise of the tools, kit or apparatus in which the system will be developed in. The case tools and programming languages that will be used to produce the final system application will have to be indicated when an application design is constructed. The design has to include the inputs and outputs that will be implemented through the system, while considering the interface which the user will interact with.

An Application architecture includes the capabilities of the system based on the client's requirements that is the application functionalities e.g.

- Managing the customer information
- Managing the warehousing system
- Processing orders and Taking orders
- Management functionalities

In Simple terms application architecture is plan where structured solutions are defined in terms of optimizing common quality attributes like security, manageability and performance while its operational and technical requirements are met.

By: LT Thwala

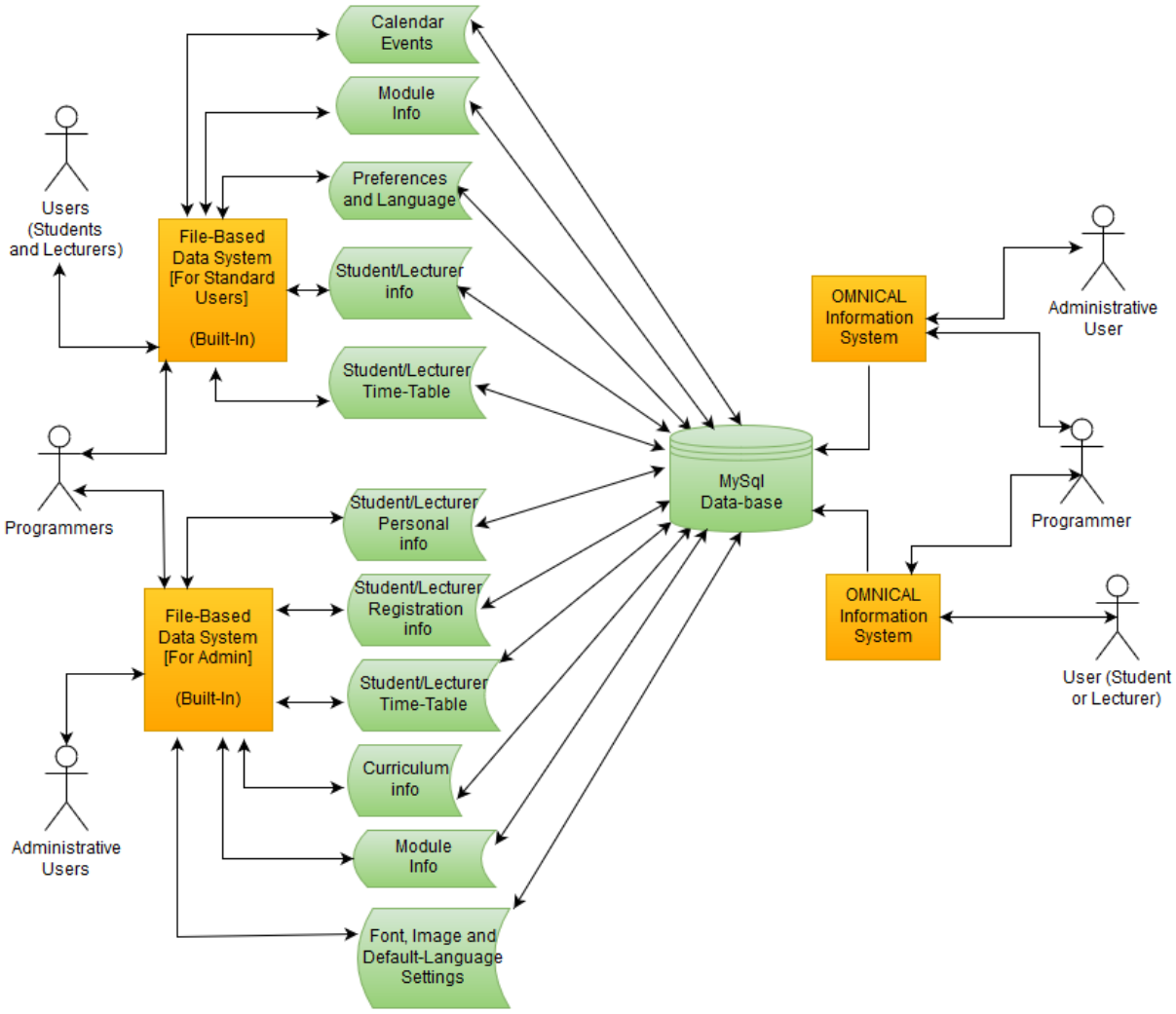
Checklist

Question	**Answer**
Will the system be centralized or decentralized?	Centralized
Will the database be distributed?	Yes
What are the user and system interface alternatives?	The system interface alternatives depend on our user type: - Student (IT or Other) - Staff - Admin
What will the software development environment be?	C#
Indicate people/computer boundaries	- Only IT students can automatically set up their timetable

By: J Muller

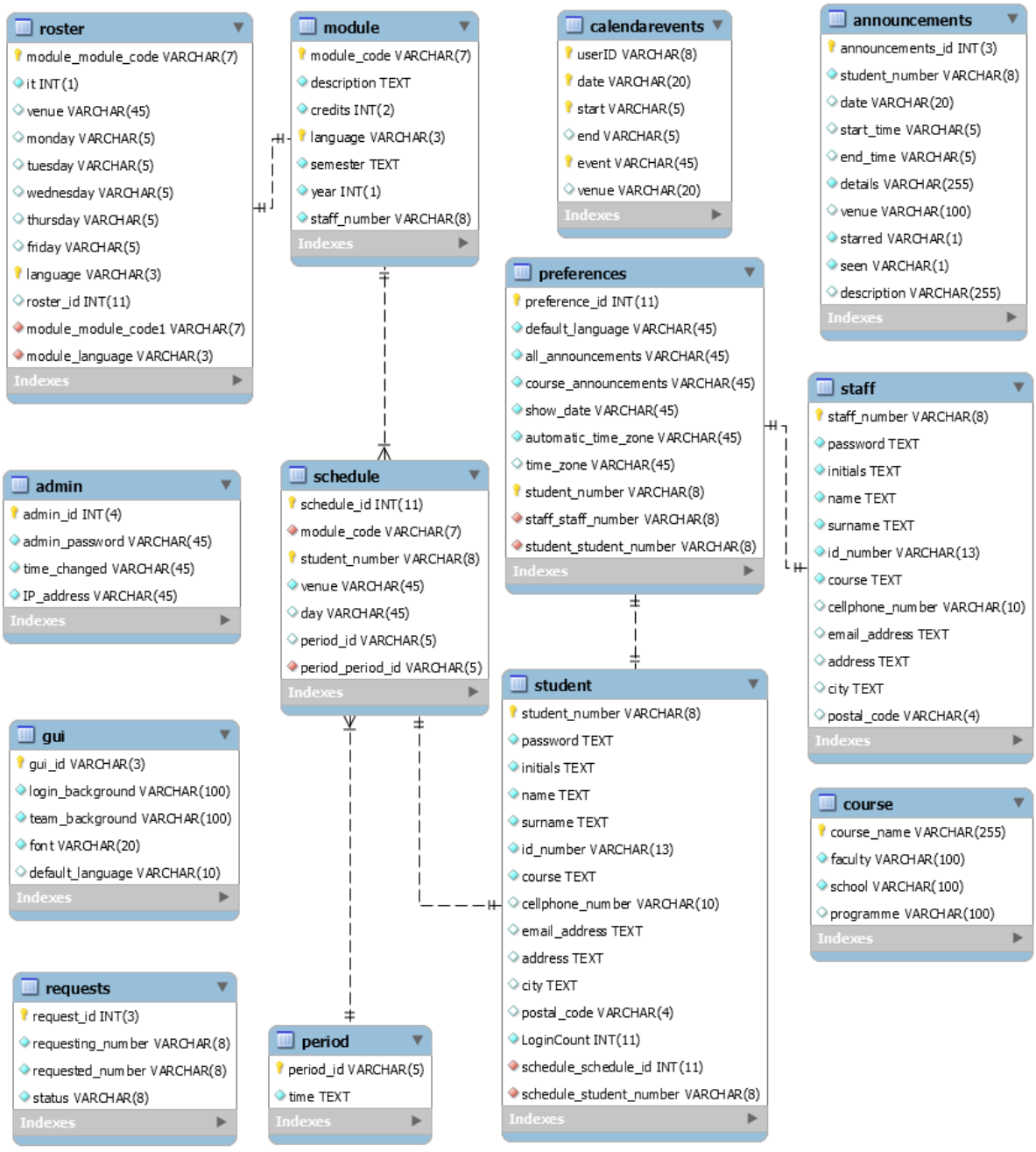
DATABASE DESIGN

Modern Data Architecture



By: MC Erasmus and LT Thwala

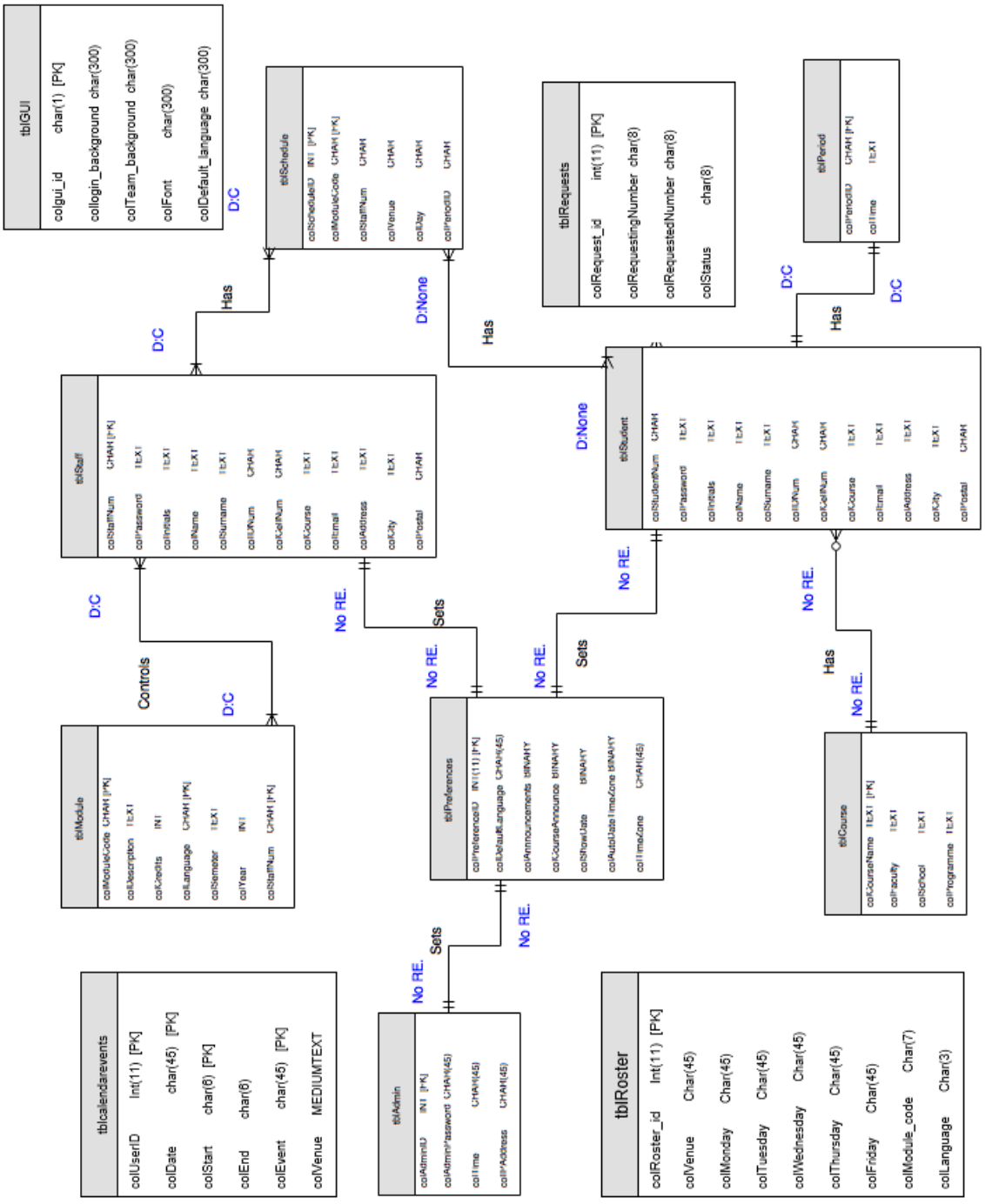
Logical Data Model



By: J Muller

SECOND SEMESTER FINAL DOCUMENTATION

Database Schema



By: TB Bensch

Database Capacity Planning

Steps:

1) *Sum the field sizes*

module total characters: 181
- module_code = 7
- description = 150
- credits = 2
- languages = 3
- semester = 10
- year = 1
- staff_number = 8

staff total characters: 465
- staff_number = 8
- password = 45
- initials = 10
- name = 45
- surname = 45
- id_number = 13
- course = 150
- cellphone_number = 10
- email_address = 45
- address = 45
- city = 45
- postal code = 4

preferences total characters: 105
- preference_id = 11
- default_language = 45
- all_announcements = 1
- course_announcements = 1
- show_dates = 1
- automatic_time_zone = 1
- time_zone = 45

admin total character: 139
- admin_id = 4
- admin_password = 45
- time_changed = 45
- IP_address = 45

course total characters: 555
- course_name = 255
- faculty = 100
- school = 100
- programme = 100

student total characters: 465
- student_number = 8
- password = 45
- initials = 10
- name = 45

- surname = 45
- id_number = 13
- course = 150
- cellphone_number = 10
- email_address = 45
- address = 45
- city = 45
- postal code = 4

schedule total characters: 121
- schedule_id = 11
- module_code = 7
- staff_number = 8
- venue = 45
- day = 45
- period_id = 5

period total characters: 20
- period_id = 5
- time = 15

roster total characters: 96
- roster_id = 11
- module_code = 7
- venue = 45
- Monday = 5
- tuesday = 5
- wednesday = 5
- thursday = 5
- friday = 5
- staff_number = 8

announcements total characters: 655
- announcement_id = 3
- student_number = 8
- date = 20
- start_time = 5
- end_time = 5
- details = 255
- venue = 100
- starred = 1
- seen = 1
- description = 255

calendarevents total characters: 78
- userID = 8
- date = 20
- start = 5
- end = 5
- event = 20
- venue = 20

gui total characters: 233
- gui_id = 3
- login_background = 100
- team_background = 100
- font = 20
- default_language = 10

requests total characters: 27
- request_id = 3
- requesting_number = 8
- requested_number = 8
- status = 8

Record size: 3 140

2) Record size x entity instances in table (using growth over 3 years)
Growth = 1.4 * 1.4 * 1.4 = 2.744

module: 181 * 40 * 2.744 = 19 866.56
staff: 465 * 50 * 2.744 = 63 798
preferences: 105 * 50 * 2.744 = 14 406
admin: 139 * 20 * 2.744 = 7 628.32
course: 555 * 100 * 2.744 = 152 292
student: 465 * 1000 * 2.744 = 1 275 960
schedule: 121 * 1000 * 2.744 = 332 024
roster: 96 * 1000 * 2.744 = 263 424
period: 20 * 10 * 2.744 = 548.80
announcements: 655 * 10000 * 2.744 = 17 973 200
calendarevents: 78 * 10000 * 2.744 = 2 140 320
gui: 233 * 2 * 2.744 = 1 278.70
requests: 27 * 1000 * 2.744 = 74 088

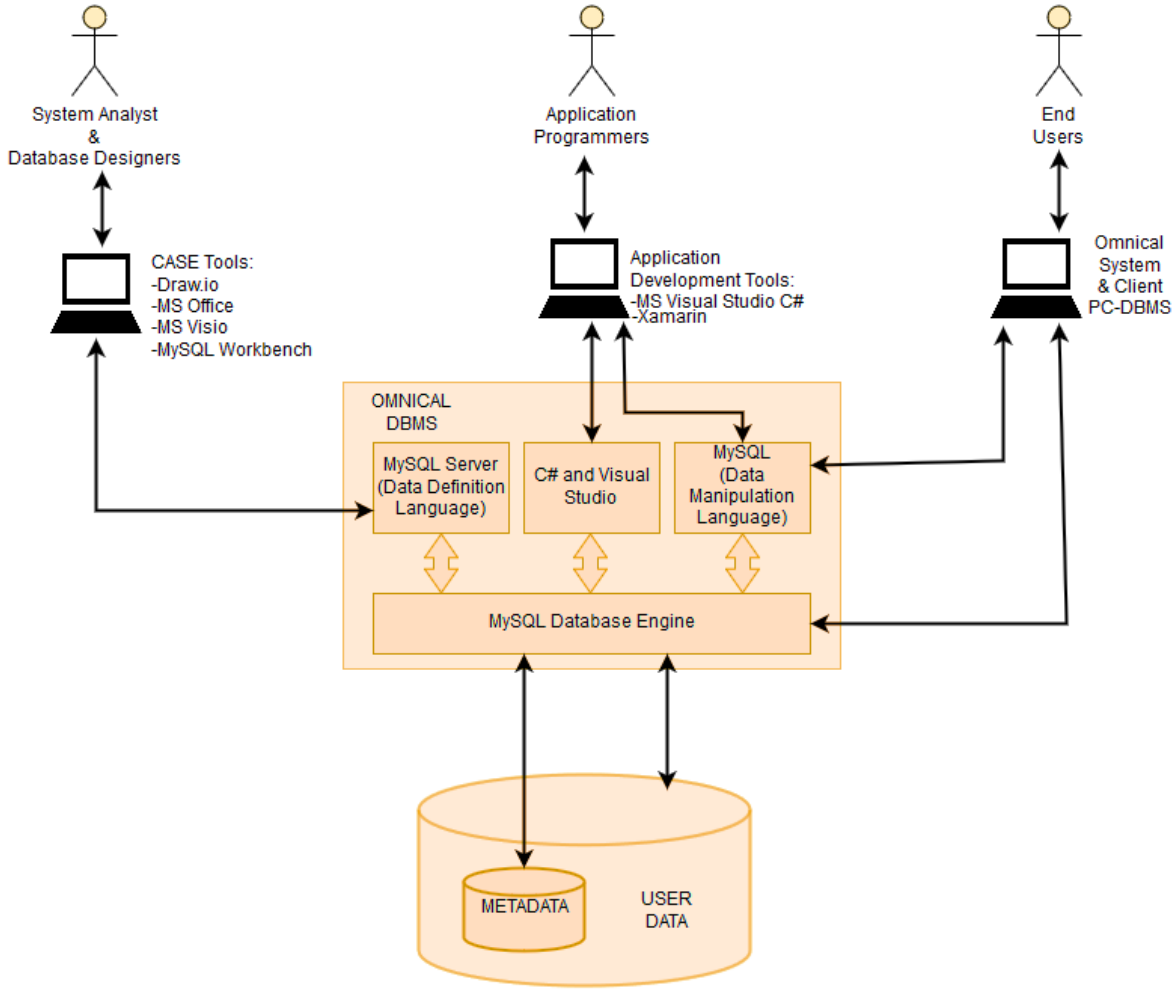
3) *Sum the table sizes*
Total = 22 261 415.38

4) *Add slack capacity buffer (10%)*
22 261 415.38 * 0.10 = 2 226 141.54

Anticipated database capacity: 2.23 GB

By: J Muller

Database Management System Architecture



By: MC Erasmus

OUTPUT DESIGN

Taxonomy for computer generated outputs

Definition Delivery	Internal output	Turnaround Output	External Output
Printer	Summary, detailed and/or exception report will be printed from the PDF file or from the program directly for internal use, in terms of diagnostic reports.	Information will be printed from the PDF file or from the program directly for reference, in terms of diagnostic reports.	Information will be printed as a hard copy. An example would be the user's timetable
Screen	Detailed, summary and exception information will be outputted onto the monitor for internal use, in terms of displaying the databases.	Information will be displayed on the monitor which could also be used as input at a later stage. In terms of databases and/ or other saved records like textfiles.	Information will be displayed on the monitor which could. For example, the user's timetable (out of editor).
Multimedia	Summary report will be created and stored in a PDF format for internal use, in terms of diagnostic reports.	Information will be stored in PDF format for reference, in terms of diagnostic reports.	Information will be created and stored in a PDF format. An example would be the user's timetable
Hyperlinks	Not applicable	Not applicable	Connects users to the Omnical Facebook and Twitter accounts

By: J Muller

Output guidelines

Guidelines	Applied
Simple to read and interpret	✓
Title for every output	✓
Time stamp every output	-
Reports and screens should include sections and headings to segment information	✓
Form base output - clearly labelled fields	✓
Tabular outputs - clearly ladled columns	✓
Reports should include legends to interpret headings	✓
Print and display only required information	✓
No manually editable information	-
Evenly spread output	✓
Easy to edit/remove or find output	✓
Computer jargon and error messages should be omitted from all outputs	✓
Output information must reach recipients while the information is pertinent	✓
The distribution of computer outputs must be sufficient to assist all users	✓

By: R Du Plooy & J Muller

Output Design Process

Step 1: Identify Outputs

Using the Physical Data Flow diagram and other documentation, the following outputs have been identified:

- Timetable (in a .PDF format)
- Timetable (in a screen output)
- Timetable (printable output)
- Announcements (in a screen output)
- Module (in a screen output)
- Calendar (in a screen output)
- View all data (in a screen output)
- Cal My Friend (in a screen output)
- Statistics (in a screen output)

By: LT Thwala (Progressed on TR Villets work)

Step 2: Specify Physical Output Requirements

Based on the nature of each output, the following Physical Requirements will need to be met:

PDF timetable:

The purpose of the output:

- Digital output, saved to the user's device, for offline or sharing purposes.

Operational, technical and economic feasibility:

- Digital file transfer that is both operationally and technically feasible. Economic feasibility is not applicable.

Omnical Timetable

26175940

	Monday	Tuesday	Wednesday	Thursday	Friday
1-2	BMAN222				ITRW225
3-4	BMAN222				ITRW225
5-6	ITRW222	WISN223	BMAN222		BMAN222
7-8	BMAN222			BMAN222	ITRW222
9-10	11-12	BMAN222			

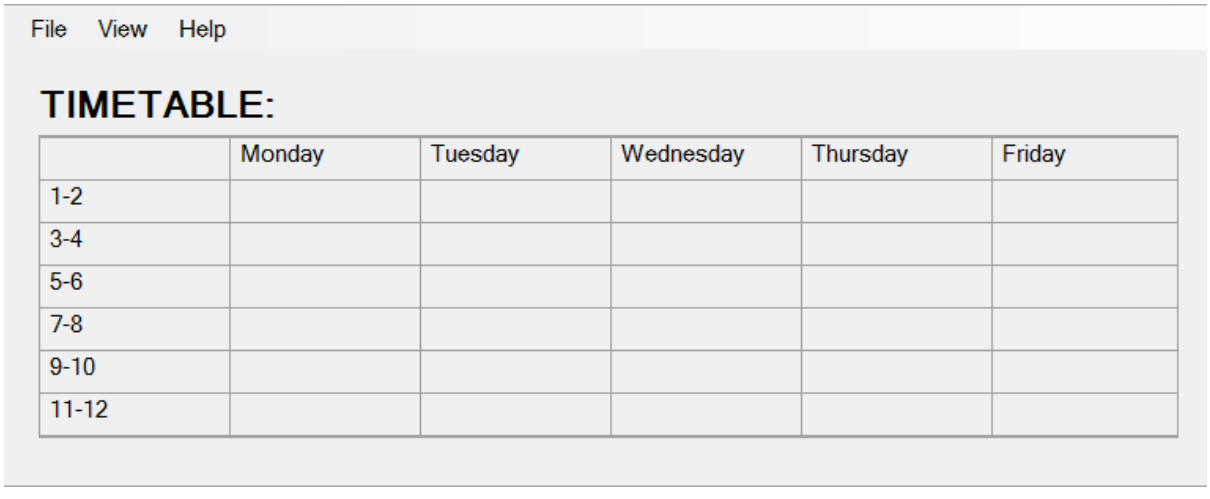
Screen Output Timetable:

The purpose of the output:

- Electronic output, shows the user their timetable on the screen.

Operational, technical and economic feasibility:

- Electronic output that is both operationally and technically feasible. Economic feasibility is not applicable. Screen output can be output on most, if not all, standard display resolutions.



Printed Timetable:

The purpose of the output:

- Physically printed output, for situations in which the user does not have access to any electronic devices and requires their timetable.

Operational, technical and economic feasibility:

- Physical output that is reasonably operationally and technically feasible, as some users or devices may not have access to a printer. Printed output is economically feasible as printing in such small quantities is inexpensive, and timetables do not change very often. Printing requires only one standard A4 sheet of paper.

Omnical Timetable

26175940

	Monday	Tuesday	Wednesday	Thursday	Friday
1-2	BMAN222				ITRW225
3-4	BMAN222				ITRW225
5-6	ITRW222	WISN223	BMAN222		BMAN222
7-8	BMAN222			BMAN222	ITRW222
9-10	11-12	BMAN222			

Screen output Module:

The purpose of the output:

- Electronic output, shows user module.

Operational, technical and economic feasibility:

- Electronic output that is both operationally and technically feasible. Economic feasibility is not applicable. Screen output can be output on most, if not all, standard display resolutions.

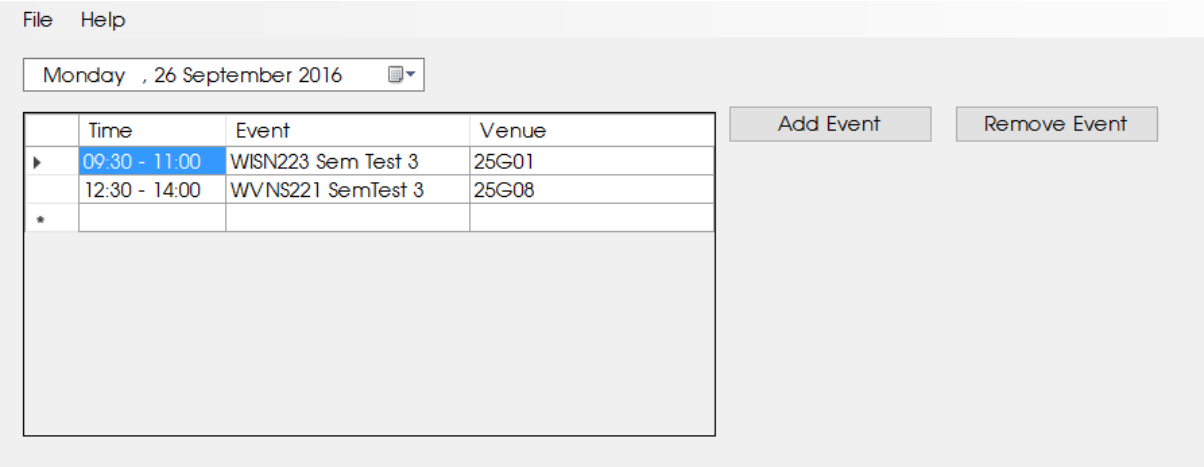
module_code	description	credits	language	semester	year	staff_numbe
ACCF111	FINANCIAL ...	16	ENG	1	1	12345678
ACCF121	FINANCIAL ...	16	ENG	2	1	TBC
ACCS111	FINANCIAL ...	16	ENG	1	1	TBC
ACCS121	FINANCIAL ...	16	ENG	2	1	TBC
AGLE111	INTRODUC...	0	ENG	1	1	TBC
AGLE121	ACADEMIC...	12	ENG	2	1	12345678
BMAN111	INTRODUC...	12	ENG	1	1	12345678
BMAN222	ENTREPR...	16	ENG	2	2	TBC
ITRW112	INTRODUC...	12	ENG	1	1	TBC

Screen output Calendar:
The purpose of the output:

- Electronic output, shows user calendar.

Operational, technical and economic feasibility:

- Electronic output that is both operationally and technically feasible. Economic feasibility is not applicable. Screen output can be output on most, if not all, standard display resolutions.

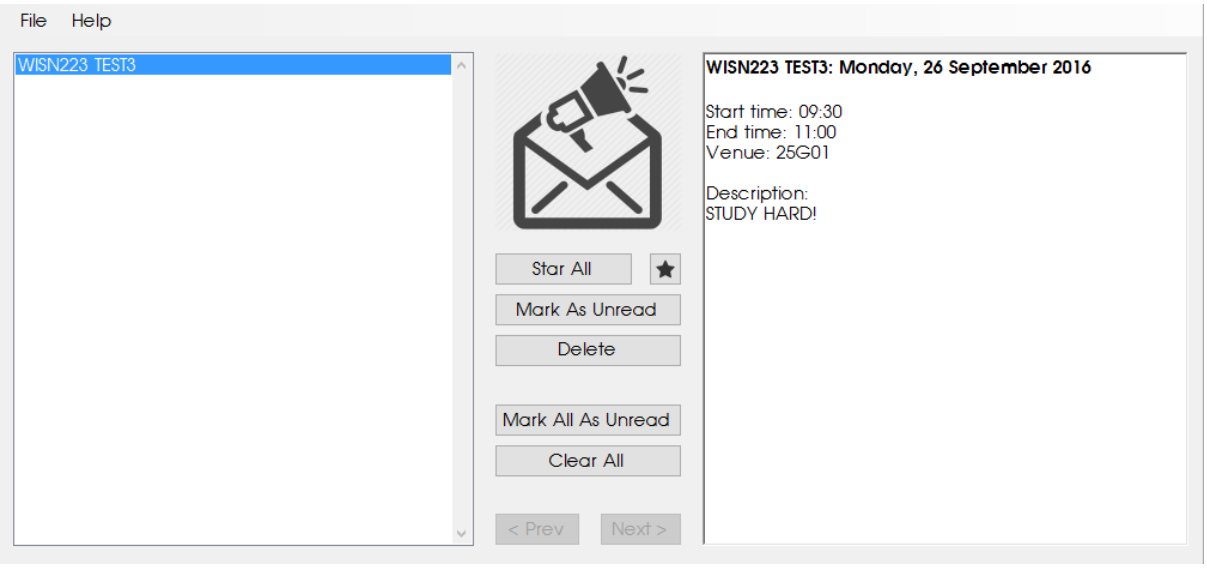


Screen output Announcements:
The purpose of the output:

- Electronic output, shows user their announcements saved from frmCalendar.

Operational, technical and economic feasibility:

- Electronic output that is both operationally and technically feasible. Economic feasibility is not applicable. Screen output can be output on most, if not all, standard display resolutions.



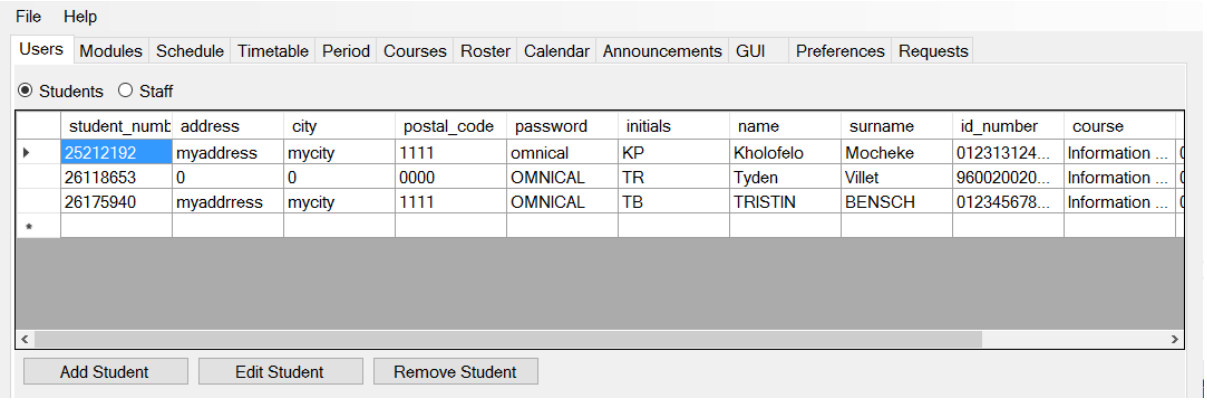
Screen output View All Data:

The purpose of the output:

- Electronic output, shows admin all data that is saved in the database.

Operational, technical and economic feasibility:

- Electronic output that is both operationally and technically feasible. Economic feasibility is not applicable. Screen output can be output on most, if not all, standard display resolutions.



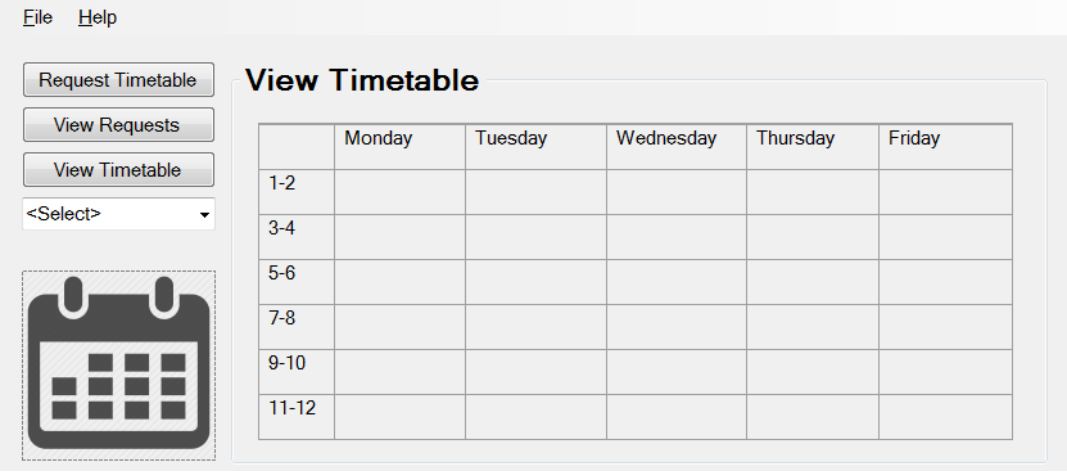
Screen Output Cal My Friend

The purpose of the output:

- The call my friend form allows a student to ask for a friend's timetable in order to schedule an appointment with the friend. Sometimes we have group activities, it could be assignments which need to be completed in pairs and somehow you both need to meet up at certain time and complete the task. To avoid any complications, a friend can use the call my friend ability of OmniCal to request a friend form

Operational, technical and economic feasibility:

- Electronic output that is both operationally and technically feasible. Economic feasibility is not applicable. Screen output can be output on most, if not all, standard display resolutions.



Screen Output Meet the Team form
The purpose of the output:

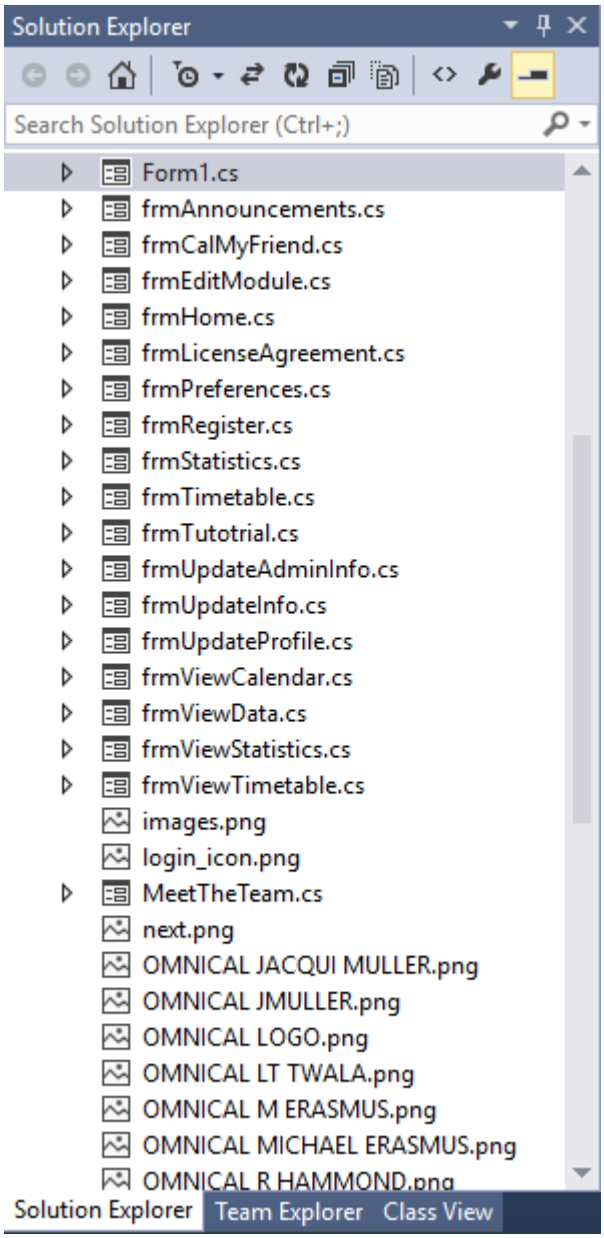
- If users require any information on the OmniCal team, this is the form they will be directed to when they click on the links on the home screen

Operational, technical and economic feasibility:

- Electronic output that is both operationally and technically feasible. Economic feasibility is not applicable. Screen output can be output on most, if not all, standard display resolutions.

By: LT Thwala (Progressed on TR Villets work)

Step 3: Design Any Pre-printed Forms



Output forms:

- frmAnnouncements
- frmCalMyFriend
- frmHome
- frmStatistics
- frmTimetable
- frmTutorial
- frmUpdateAdmin
- frmUpdateProfile
- frmViewCalendar
- frmViewData
- frmViewStatistics
- frmViewTimetable
- MeetTheTeam

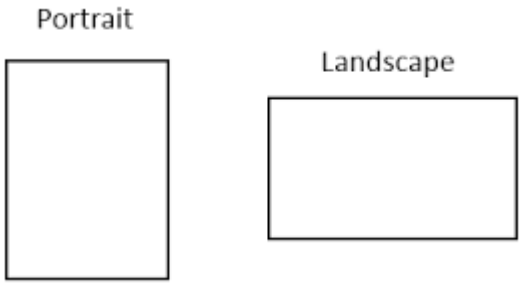
By: J Muller

Step 4: Design, Validate and test Outputs

Form Name	Designed	Validated	Output tested
frmAnnouncements	✓	✓	✓
frmCalMyFriend	✓	✓	✓
frmHome	✓	✓	✓
frmStatistics	✓	✓	✓
frmTimetable	✓	✓	✓
frmTutorial	✓	✓	✓
frmUpdateAdmin	✓	✓	✓
frmUpdateProfile	✓	✓	✓
frmViewCalendar	✓	✓	✓
frmViewData	✓	✓	✓
frmViewStatistics	✓	✓	✓
frmViewTimetable	✓	✓	✓
MeetTheTeam	✓	✓	✓

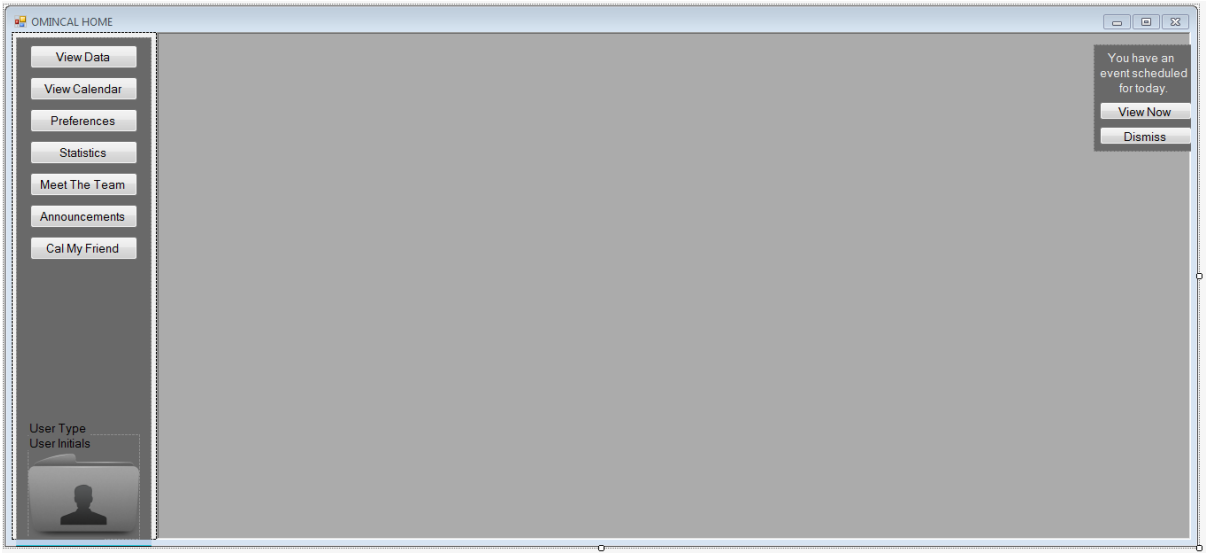
By: J Muller

The Tabular Design principles of OmniCal

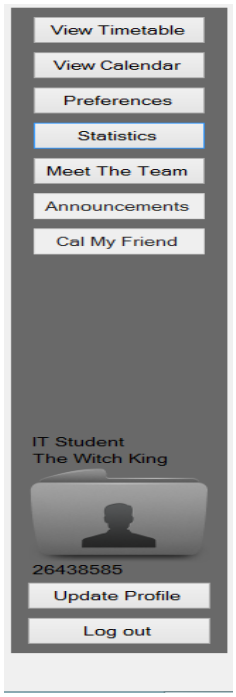
Design issue	The Guidelines for the Design	Example
The size of the page	The sizes of the pages will be according to the user requirements. The default printing size is A4.	N/A
The page orientation	The page orientation of the timetable will be in landscape format on contrary that the administrator can change it to portrait format	
The heading for the different pages	The page heading will appear on each and every page and it should include the date and time.	February 13, 2016 Page 1 of 1 Omnical Timetable Student Nr. Monday Tuesday Wednesday Thursday Friday 1-2 3-4 5-6 7-8 9-10 11-12

OmniCal Legend	The numbers on the time tables will show the session times for the specified periods attended The display form and the printed form will have this abbreviations	OmniCal Legend ENG – English AFR – Afrikaans 1-2 - 8:00 - 9:20 3-4 - 9:30 - 10:50 5-6 - 11:00 – 12:30 7-8 - 12:30 - 13:50 9-10 - 14:00 - 15:20 11-12 - 15:30 - 16:50

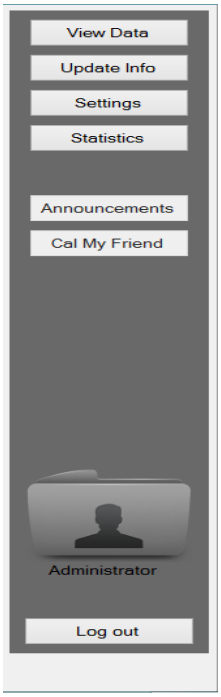
The home Page of Omnical according to each User login



The home page is the first page which will be seen by the user. Depending the user types, access will be restricted to normal student users. The user feedback is very important, and the user need not to be confused by any output design for OmniCal, hence the design is has been partitioned for different users.
Paper exercise is very strict on verification. All the input designs are taken into consideration first. Like the specifications of the system.



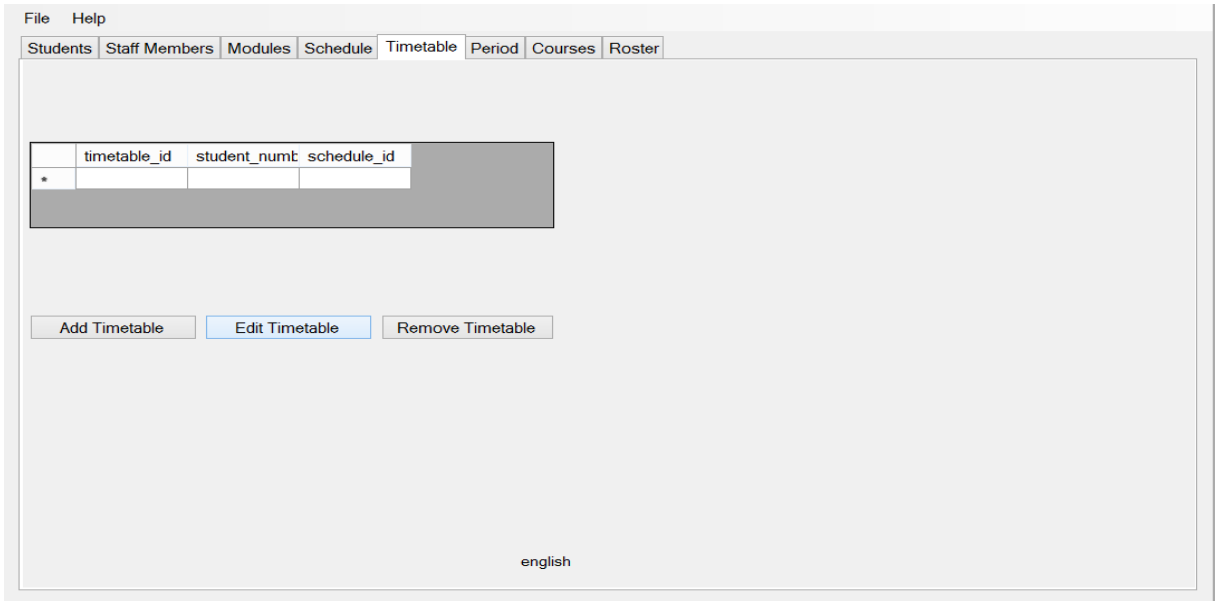
User - Student



Administration

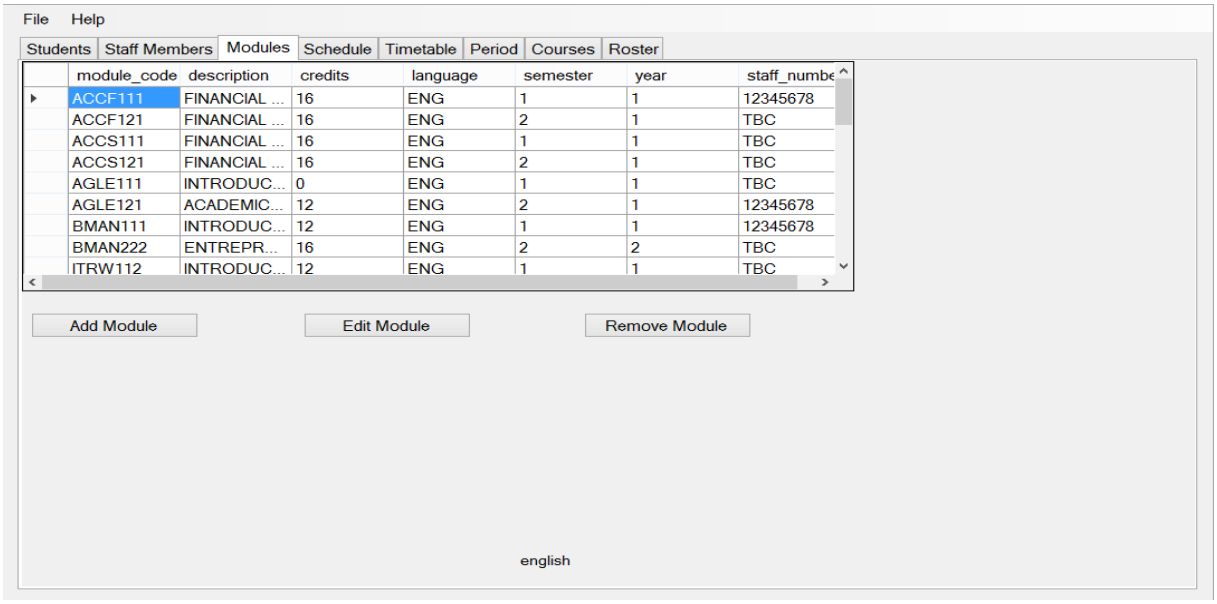
The administrator will access this page if he or she has to change any information regarding the student or staff member especially when the both users have no access to the attribute being changed as the administrator has access to change any information or attribute on the system

The View Data form (Timetable tab) of OmniCal accessible by admin only



After the information has been edited or changed by the administrator this form should display the following information on the Module page. The attributes of this page are strictly restrained to admin access only, students and lecturers cannot access this page.

The View Data form (Modules tab) of OmniCal accessible by admin only



Requirements Verifications		C /NC	
Maximum dimensions	1″* 2″ * 4″	C	Drawing

Login page – Design Validation

The registration step has to be addressed first and not missed as it has more obvious & important in all the OmniCal validation steps.

File View Help

TIMETABLE:

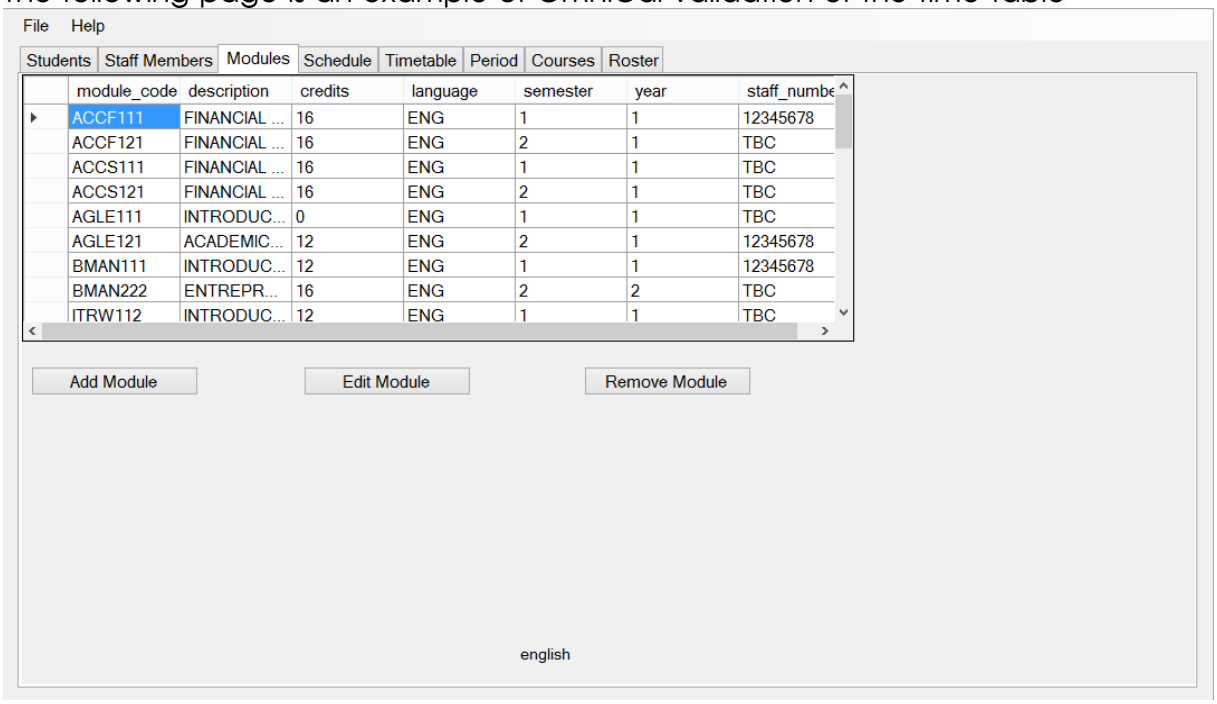
	Monday	Tuesday	Wednesday	Thursday	Friday
1-2					
3-4					
5-6					
7-8					
9-10					
11-12					

When a time table is edited manually or automatically, a version of the product is validated and the process is call the validation step. It can be considered as the first production unit but not all the time when the time table is created and we refer to the first production unit.

Validation vs Verification

Validation for OmniCal is the step that place you to build a model of the timetable, and would be completed against the necessities as modified after verification. This does no longer necessarily suggest the first production unit, but it can. It can also be an engineering model, which some systems use to prove the first units/events of a simple new time timetable, or it can be an element of the sketch which is different from the previous semester schedule, when the result is an amendment of an already-edited design.

The following page is an example of OmniCal validation of the time-table



INPUT DESIGN

Taxonomy for computer generated inputs

Process Method	Data capture	Data Entry	Data Processing
Keyboard	- User information entered by the user is recorded in the database which is collected during the registration. - Additional information regarding the user's timetable is acquired through the timetable creation process	Data is entered through the keyboard: - Log in - Register - Timetable details - Changes to data by admin - Editing/ Adding modules by staff	Data inputted via keyboard runs through a validation process as a key is pushed.
Mouse	- User information entered by the user is recorded in the database which is collected during the registration. - Additional information regarding the user's timetable is acquired through the timetable creation process - Navigation through the OmniCal system is dependent on this component along with GUI components such as buttons, radiobuttons, checkboxes, comboboxes, scrollbars, etc.	Data is entered through the mouse: - Log in - Register - Timetable details - Changes to data by admin - Editing/ Adding modules by staff - Navigation through OmniCal - GUI Compenents: * buttons * checboxes * radiobuttons * menus * labels	- Further validation processes of data are executed once the mouse has clicked on certain GUI components. - Once the mouse has been clicked, the data is processed and stored in the necessary and applicable data store
Touch Screen	- User information entered by the user is recorded in the database which is collected during the registration. - Additional information regarding the user's timetable is acquired through the timetable creation process - Navigation through the OmniCal system is dependent on this component along with GUI components such as buttons, radiobuttons, checkboxes, comboboxes, scrollbars, etc.	Data is entered through the mouse: - Log in - Register - Timetable details - Changes to data by admin - Editing/ Adding modules by staff - Navigation through OmniCal - GUI Compenents: * buttons * checboxes * radiobuttons * menus * labels	- Further validation processes of data are executed once the mouse has clicked on certain GUI components. - Once the mouse has been clicked, the data is processed and stored in the necessary and applicable data store

By: J Muller & MC Erasmus

Input Design Guidelines

Guidelines	Applied
Capture only variable data	✓
Do not capture data that can be calculated or stored in computer programs	✓
Use codes for appropriate attributes	✓
Include instruction to complete forms	✓
Minimize handwriting/typing	✓
Sequenced data entry (Left to right and top to bottom)	✓
Use design based on known metaphors	✓

The Input Design process

Step one - Review logical Requirements and Identify System Inputs

OmniCal Data-Structure

Data Structure Defining Logical Requirements Comments	
Timetable – Student Number	
- Date	← Different identifier for the output
- Events	← This value will execute whenever a
- Times	timetable is generated
+ Student Name	← Optional value, meaning if the student
+ Student Address	number is not used, the name could be
- Student Curriculum	used in its place

Data Requirements for input

- String Name
- String surname
- String Phone Number
- String Address
- String password

The User Interface design

To input the student data into the system, the system administrator or analyst may have to produce a source document, procedures, methods and input screens to edit the data into the system.

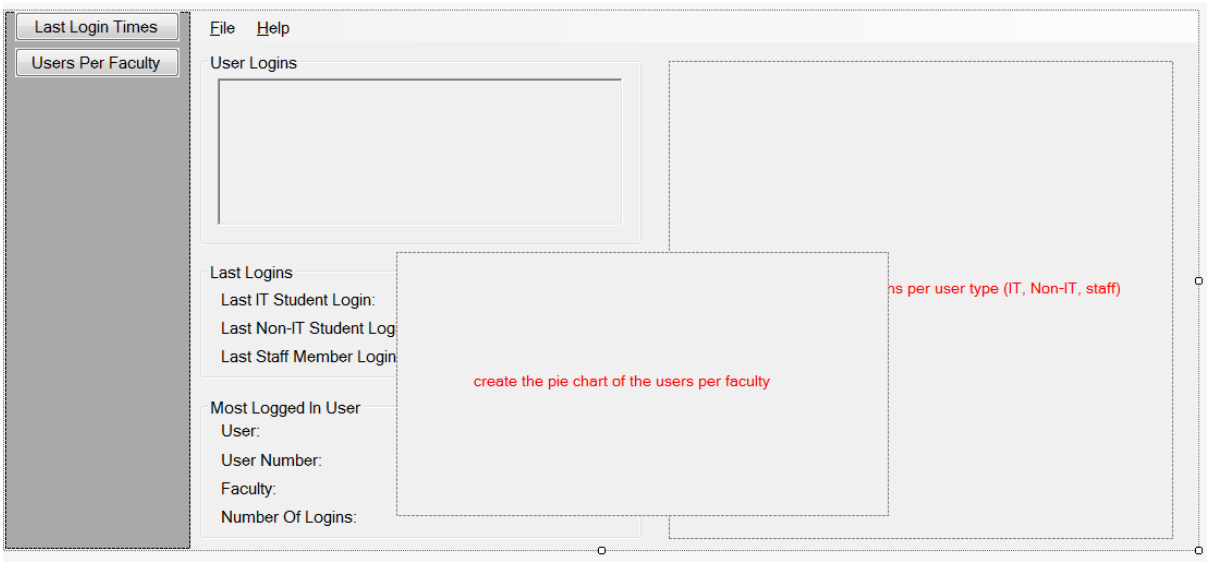
Implementation

The Omnical choice of methods or data to be captured will influence the processing of the system.

- Mouse
- Screen
- Touchscreens (Smart-phones users)
- HyperLink (When directed to the "Know the Team" Page)
- Key-Board

The view Statistic form will show the student the rate at which they utilize the app.

- How they login
- Only IT students Stats
- Even staff members can view and check how effective is the app



Step 2: Select Appropriate GUI Controls

1. Drop-down list (The user have to check their user types)
2. Textbox – The user enters their user names in this entry
3. Checkbox – If the user is not sure of his or her password, they can view their password by clicking on the checkbox
4. The attributes indicated for number 4 and 5 are for administration or the analyst to change any data that has to be edited on the system

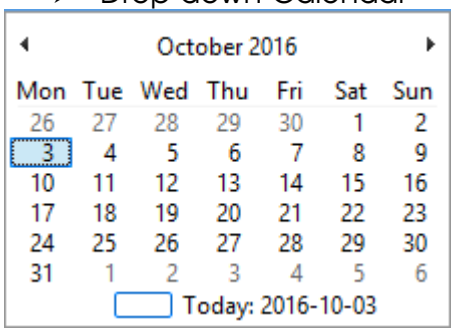
module_code	description	credits	language	semester	year	staff_numbe
ACCF1	FINANCIAL ...	16	ENG	1	1	1234
ACCF12	FINANCIAL ...	16	ENG	2	1	TB
ACCS111	FINANCIAL ...	16	ENG	1	1	TBC
ACCS121	FINANCIAL ...	16	ENG	2	1	TBC
AGLE111	INTRODUC...	0	ENG	1	1	TBC
AGLE121	ACADEMIC...	12	ENG	2	1	12345678
BMAN111	INTRODUC...	12	ENG	1	1	12345678
BMAN222	ENTREPR...	16	ENG	2	2	TBC
ITRW112	INTRODUC...	12	ENG	1	1	TBC

4

5

Other Advanced input controls

- Drop-down Calendar



- Internet hyperlink
  OmniCal has a Facebook Page, Twitter and YouTube accounts where users can enquire on any technical issue or suggest for improvements, as well as on the website.

System problems for input design:

- Inputs originate with system users, human factors play great function in input design. Inputs ought to be as simple as possible and be designed to limit the opportunity of mistaken data being entered.
- Capture solely variable data: don't enter regular information because it is probable saved in database table
- Don't seize information that can be calculated or saved in laptop programs
- Use codes for terrific attributes: codes can be translated in computer applications by using tables.

Step 3: Design, Validate, and Test inputs

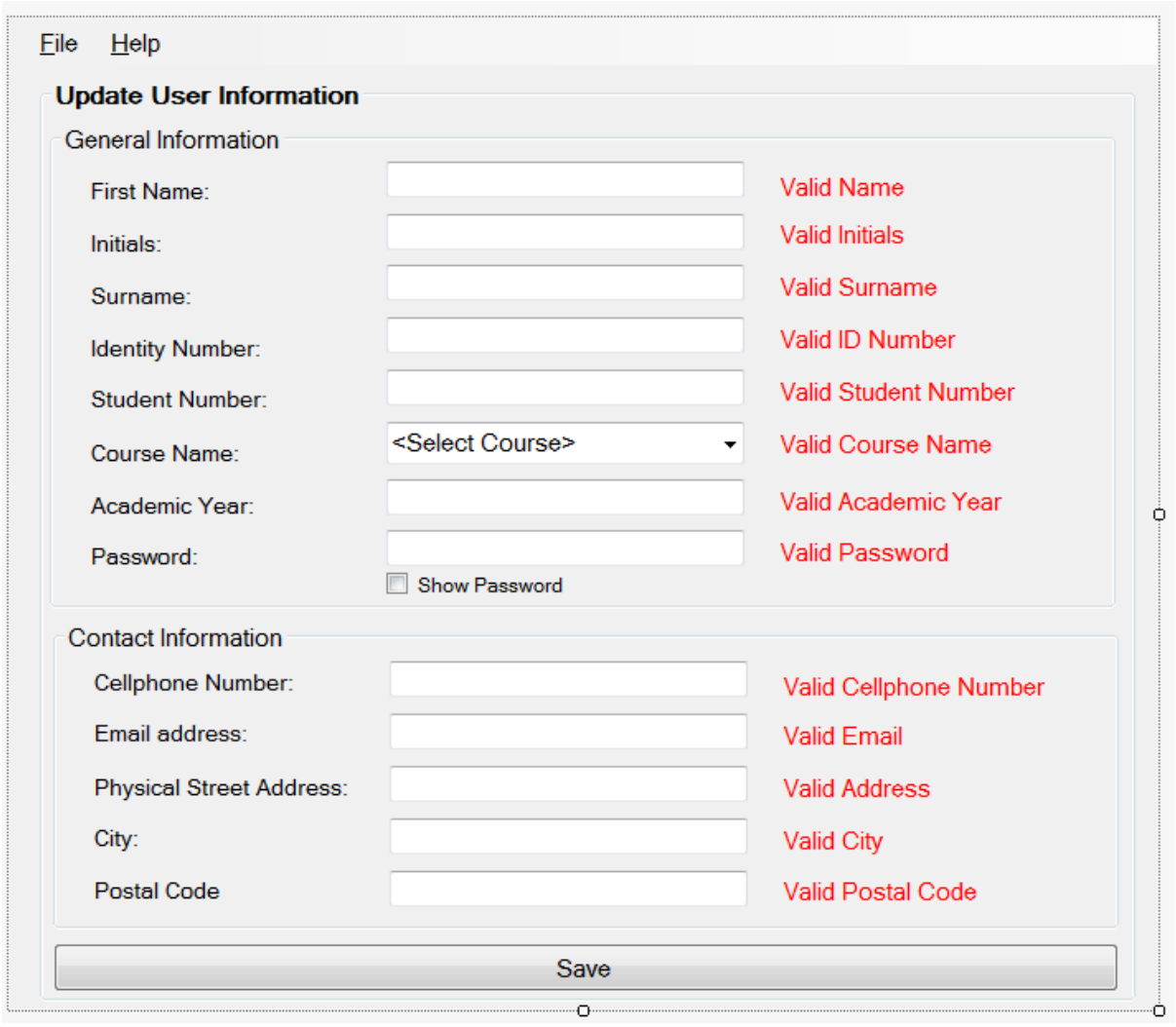
- Label – Labels are used to navigate the user to enter relevant information into the system
- Text- Boxes – The fields are for editing any information required
- Validation labels – If the user enters any irrelevant information, the entered information will not be processed and the validation text will appear to show the user how he or she should enter the values required
- Checkbox (for password visibility)
- Dropdown list (for manual selection)

Omnical Timetable

Student Nr.

	Monday	Tuesday	Wednesday	Thursday	Friday
1-2	3-4	5-6	7-8	9-10	11-12

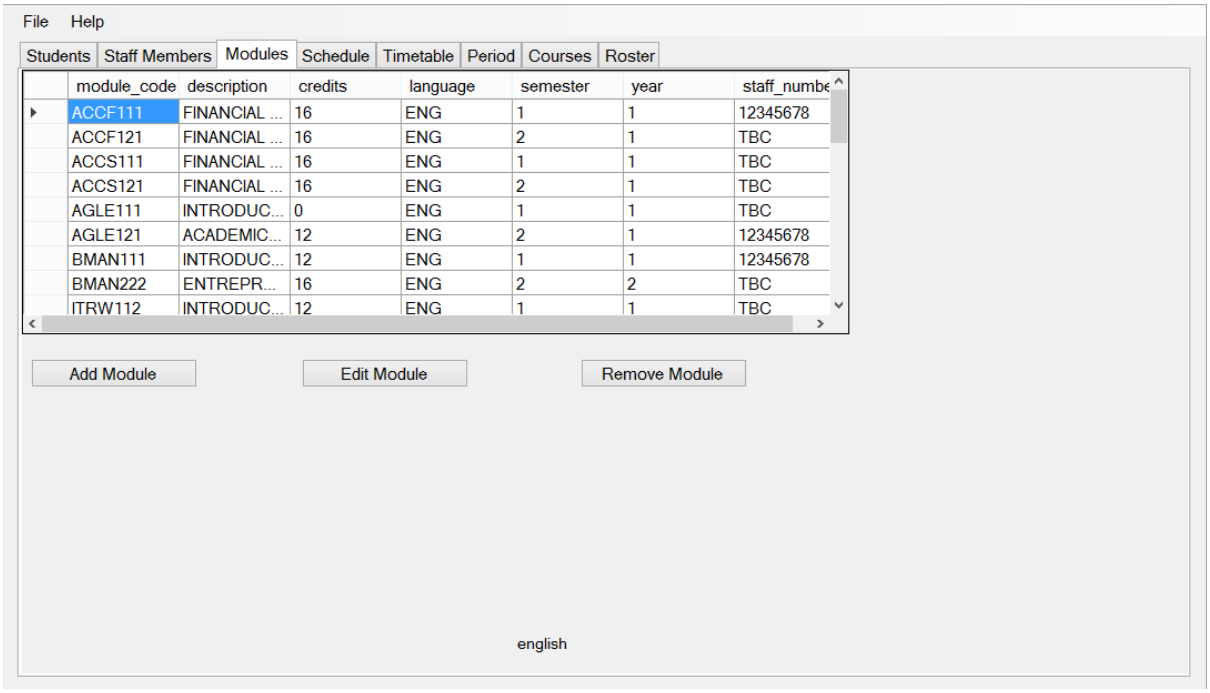
1 The student number, name, course and the modules which the students has to attend have significant role in uploading the timetable for every user. These fields are manually entered by the user and some of they are automatically generated by the system.



The update profile form will only save information if the input information has been validated first.

Step four: If necessary, design Source Documents

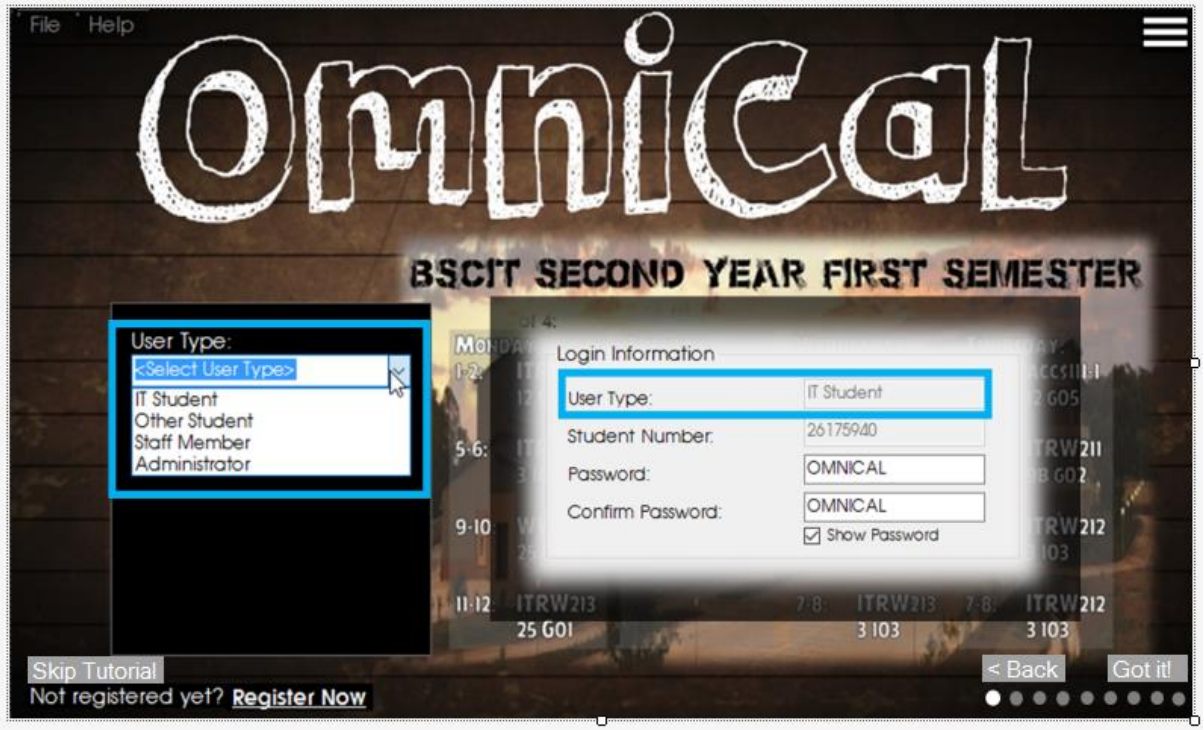
We characterize and prevent a rough out and despatch of a two way-aim overtures to-timetabling system for OmniCal System, and we assess the frugal of our principles nearby the authentic plan go off at a line which was manually constructed for the first and second semester school term. The compare with timetables for OmniCal, as at superb universities like NWU demand be created at the students reserve in brief, and our "wish list" of pairs of edify walk we would parade to give in non-overlapping timeslots is order advantaged than if we were to profit only those go wool-gathering absolutely must be in non-overlapping timeslots.



This necessitates assigning possibility levels of strife weightiness for the session or lecturer classes' slots and correcting our objective to minimize total clashes in your timetable and making sure that a students is well aware of every class they have to attend. Our put off objective is to establish timetables focus expectation in help in harmony schedules for the instructors and students.

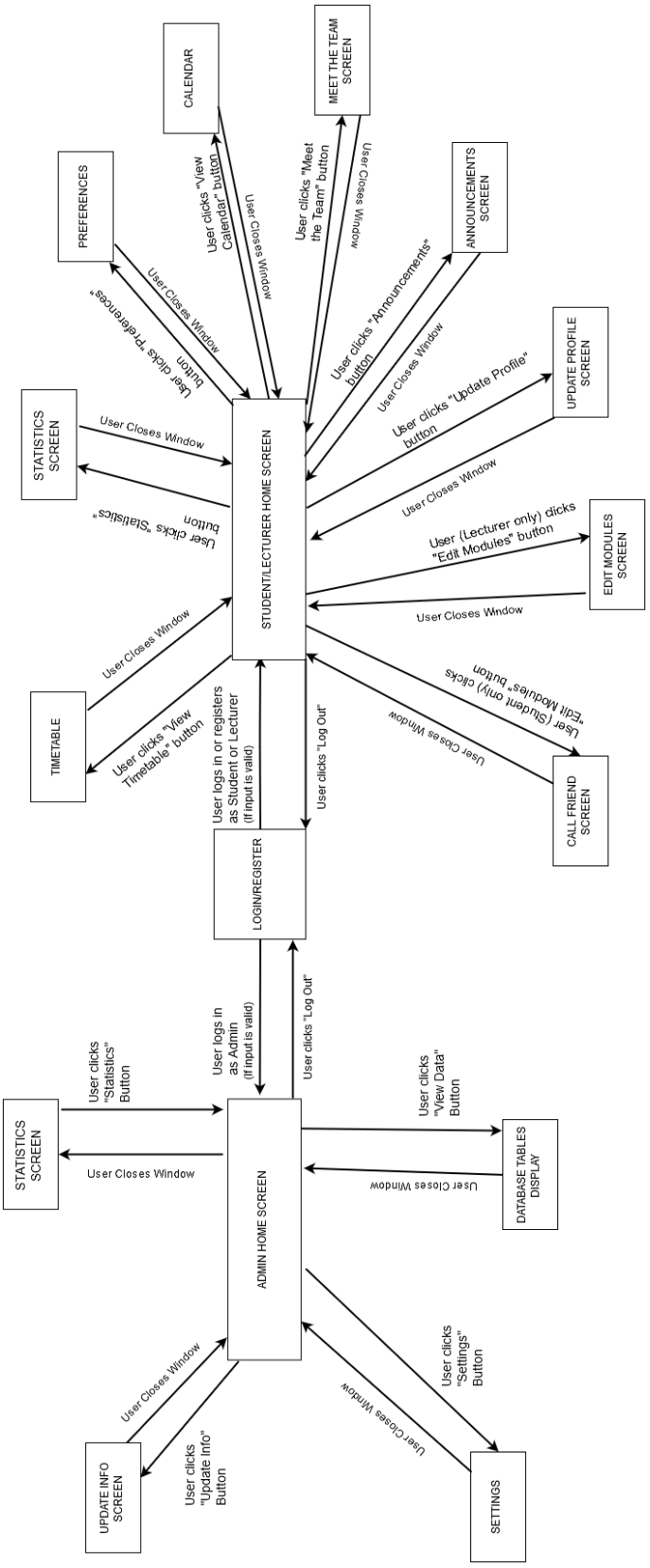
Meet the team

If the user doesn't feel like the system is not working to the needs of the students by providing unreliable timetables. It is possible that the school original time-table is not scheduled well and therefore OmniCal cannot be blamed for those errors. If there's any technical issue that has to be attended to a user can contact the OmniCal team through the website.



The most important process in education institution is scheduling the exam time table. You must be able to satisfy a set of certain constraints when setting or scheduling time tables for institutions. OmniCal presents a tutorial form for setting up a time table for individual students

Transition Diagram



By: MC Erasmus

SECOND SEMESTER FINAL DOCUMENTATION

GRAPHICAL USER INTERFACE DESIGN

GUI Design Guidelines

Guidelines	Applied
Tell the user what the system expects from them	✓
Tell user that data has been entered correctly	✓
Explain the reason way there is a delay in processing	✓
Tell a user that a task was completed or was not completed	✓
Error messages should be displayed in the same area	✓
Messages should be display long enough to be read by the user	✓
Use display attributes sparingly	✓
Default values for user should be specified	-
Anticipate the error users might make	✓
Errors should be fixed before moving on	✓
In case of a system crash, help should be contacted	✓

By: R Du Plooy

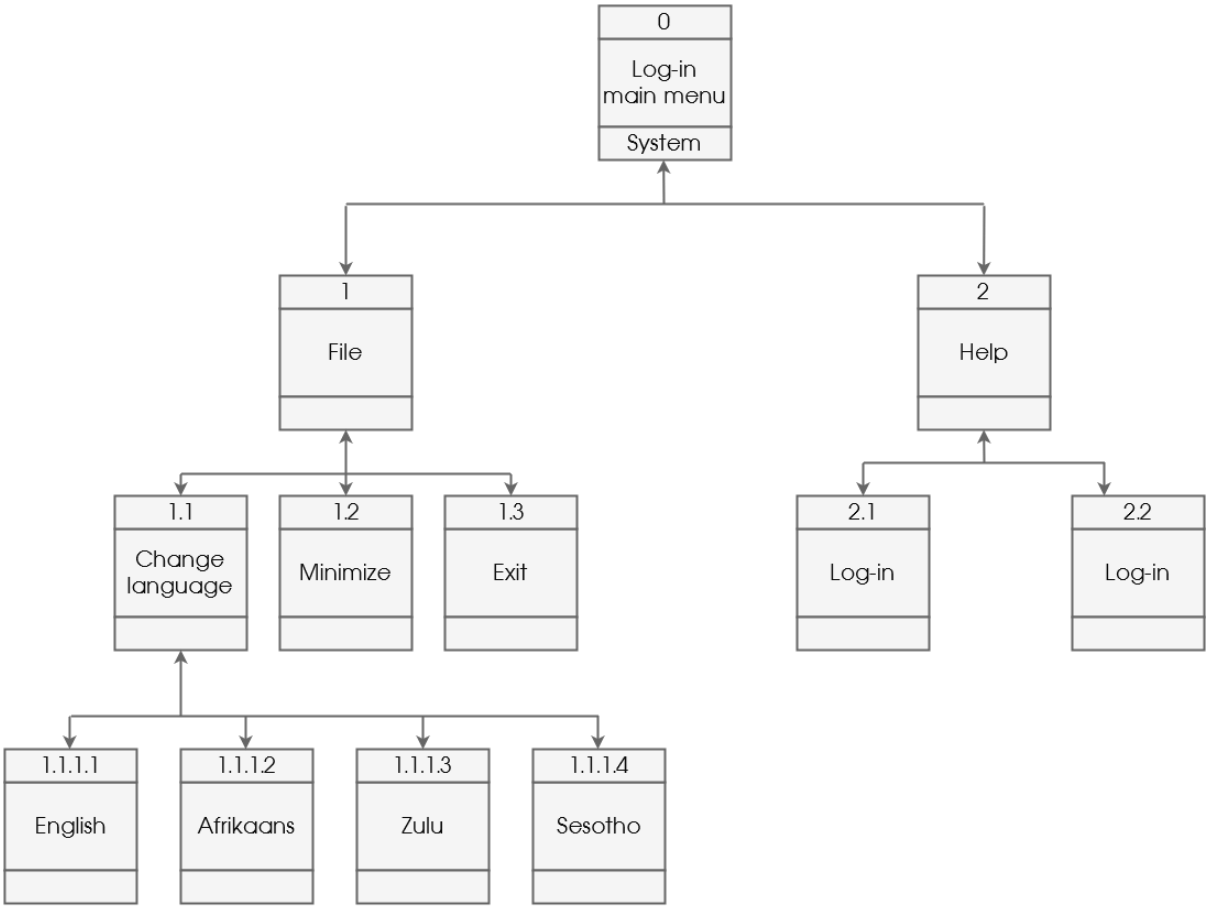
Dialog Chart Checklist

Unique menu options	
Log in	
Home screen	
View Timetable	
Meet the Team	
Main menus that are identical	
View calendar	
Preferences	
Announcements	
Statistics	
Update Profile	
Cal Friend	

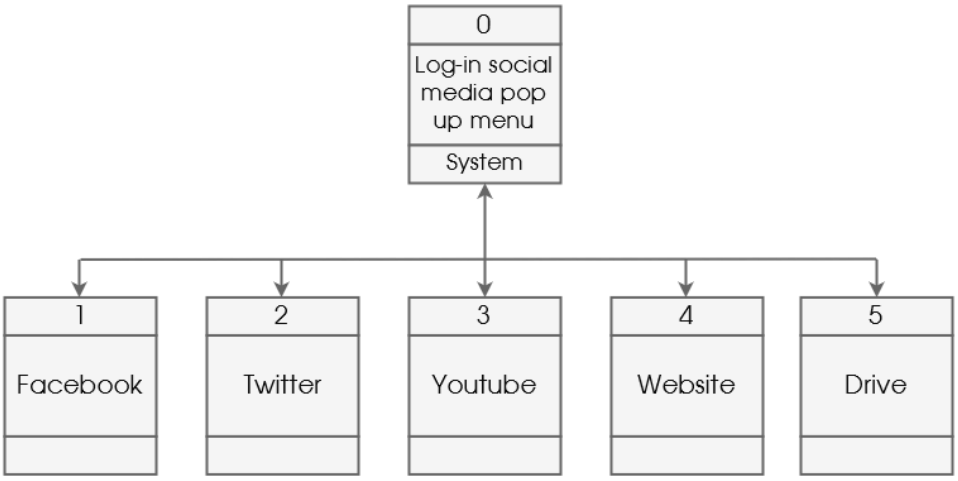
By: R Du Plooy

Dialog Charts

Login Main Menu

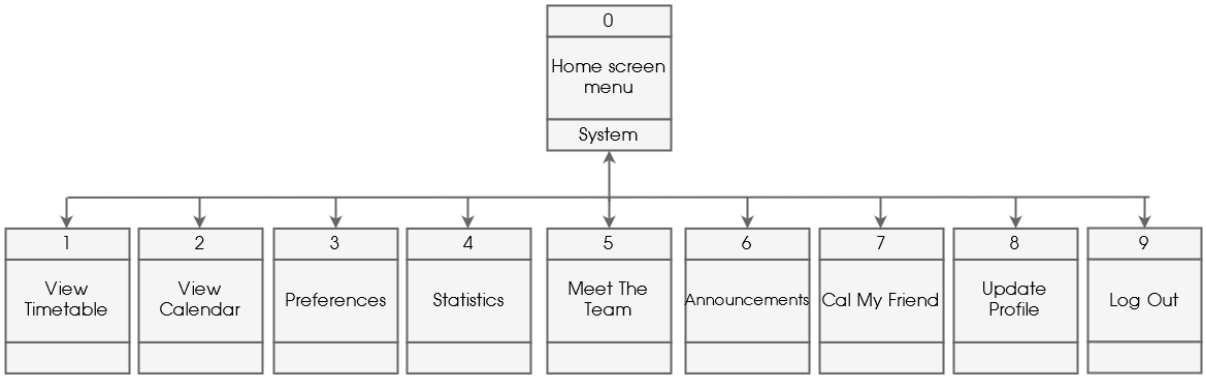


Login Social Media Pop up Menu

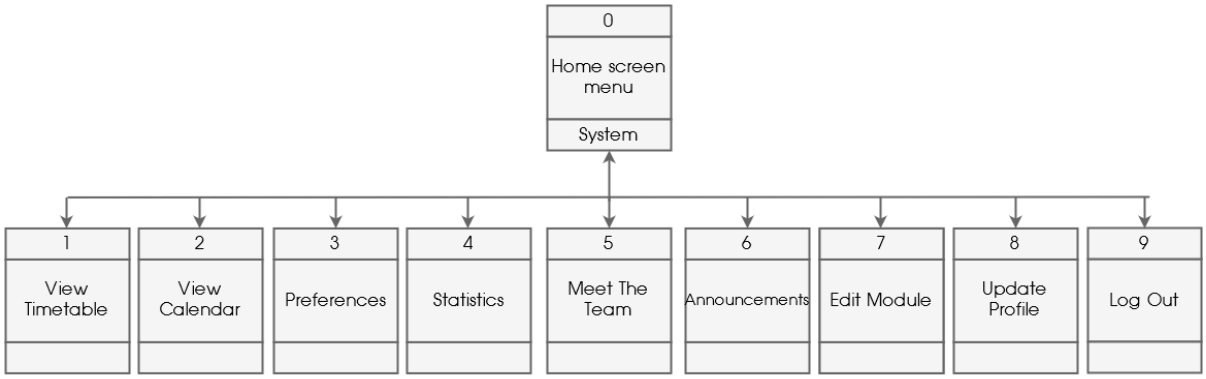


Home Screen Menu

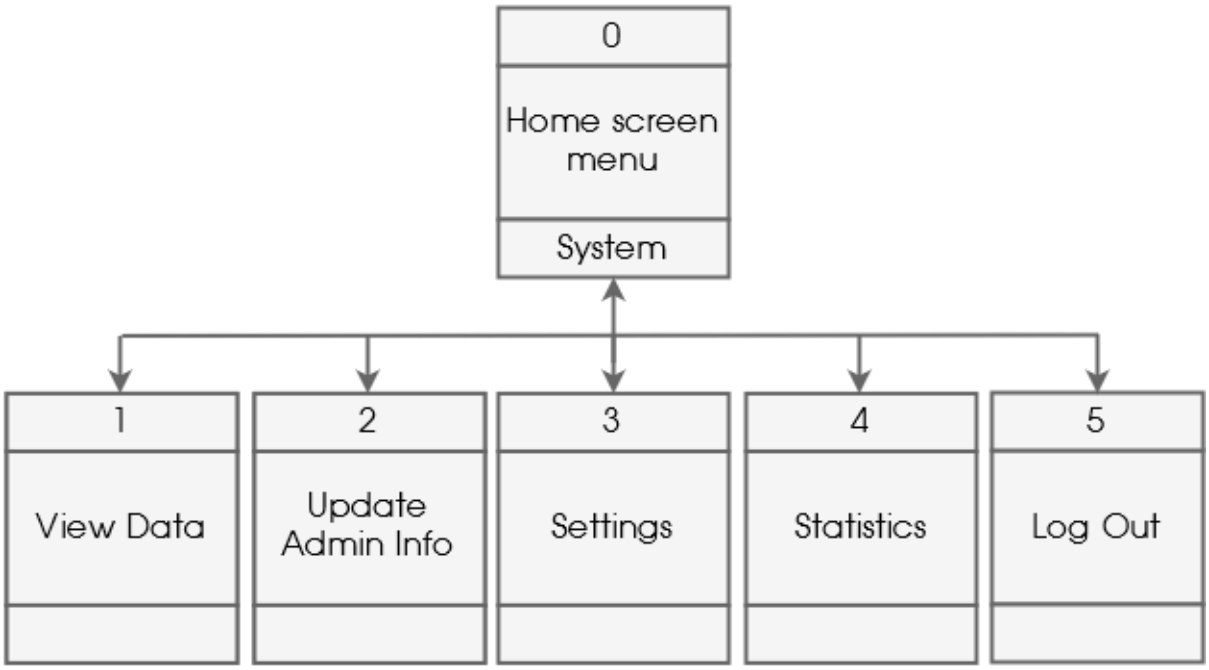
Student



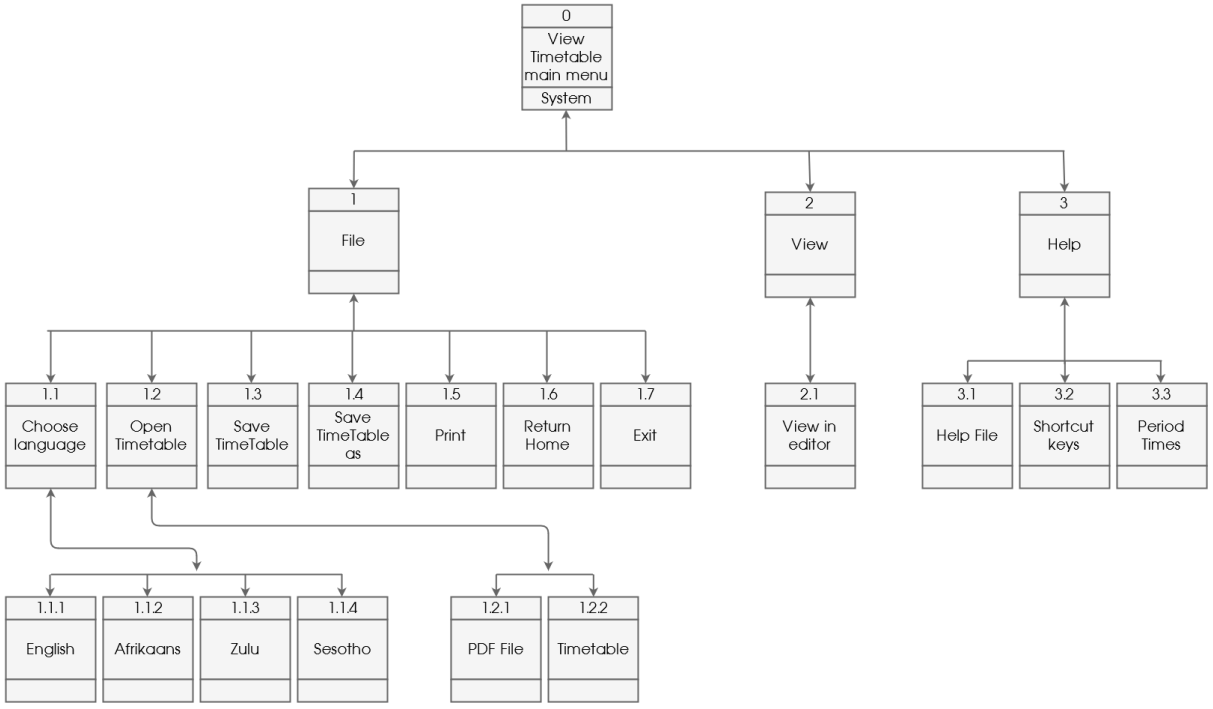
Staff



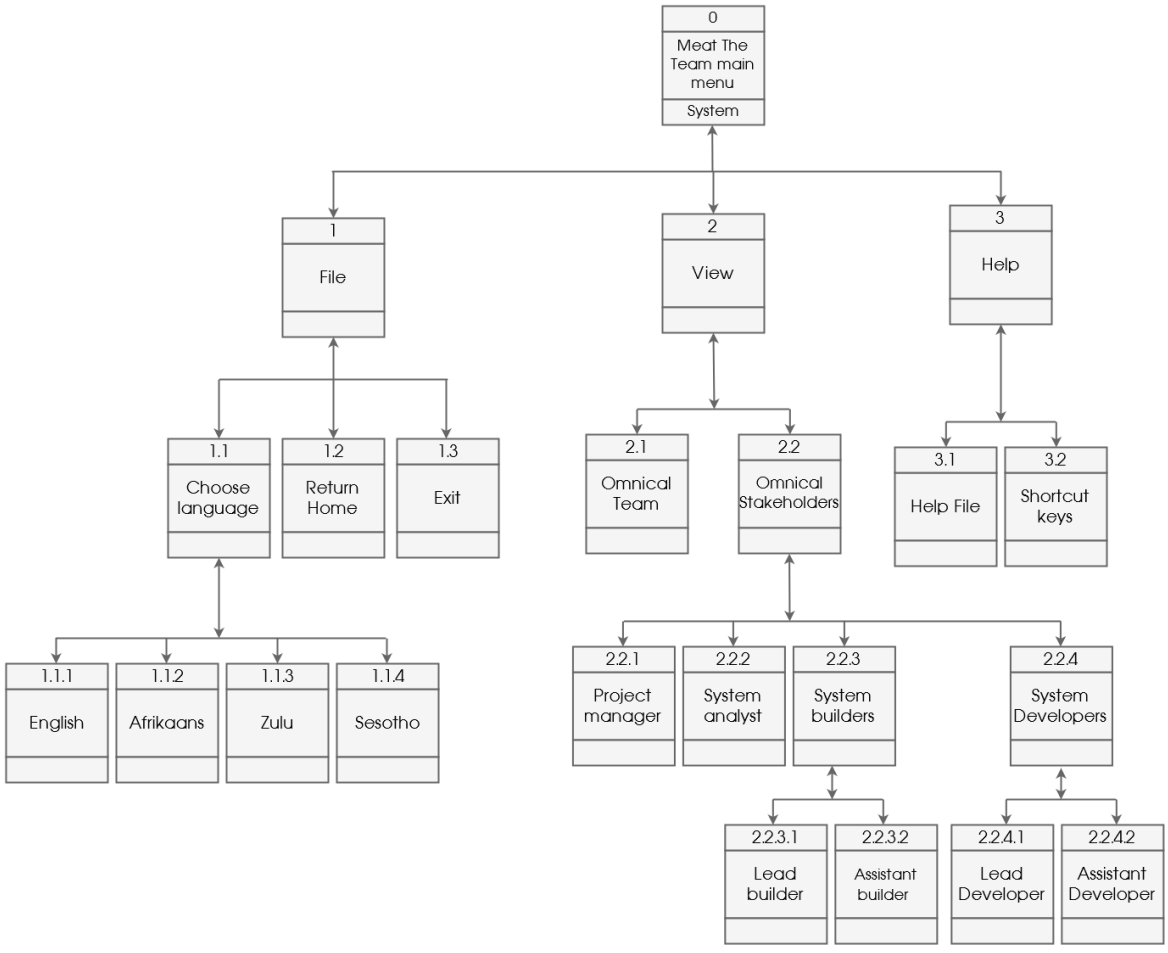
Admin



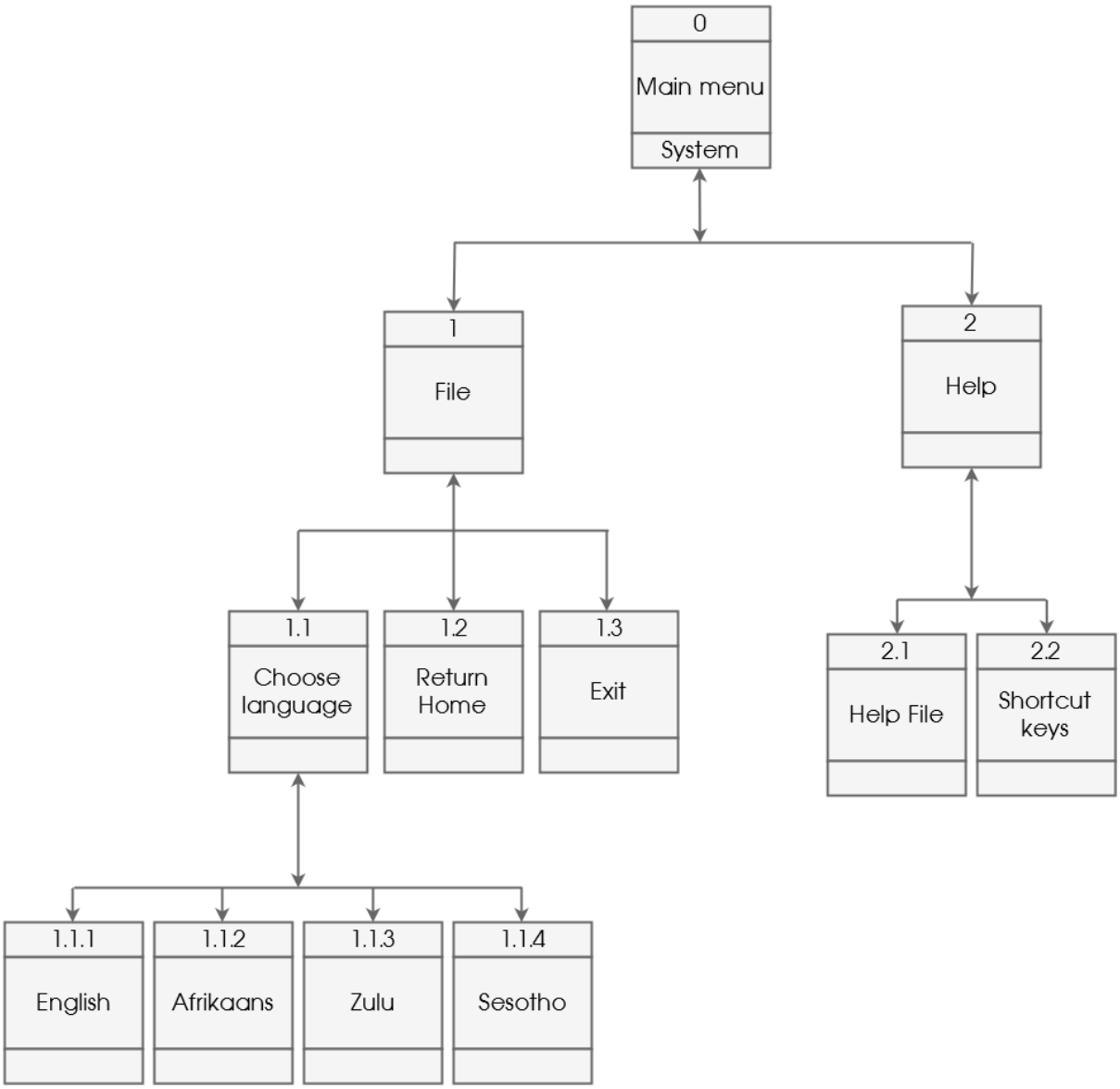
View Timetable Main Menu



Meet The Team Main Menu



Custom Form Main Menu



By: R Du Plooy & J Muller

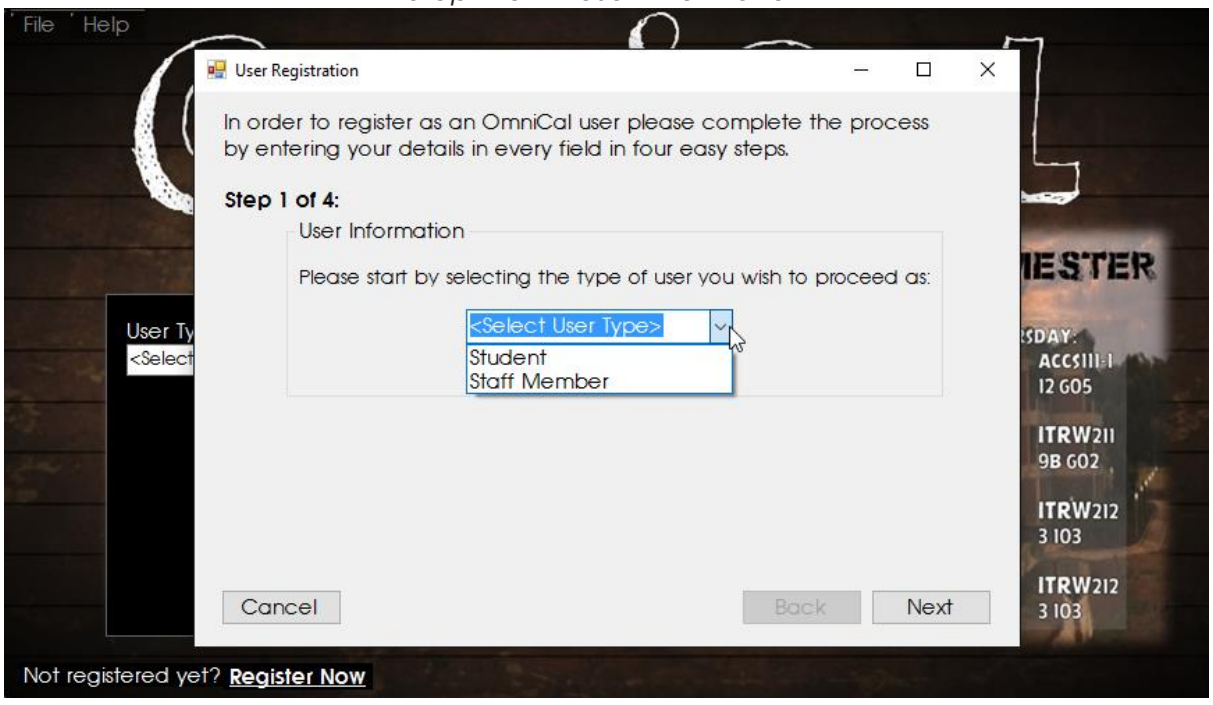
Revised System Input screens

Log in Screen: Allows for user input log in information

By: J Muller

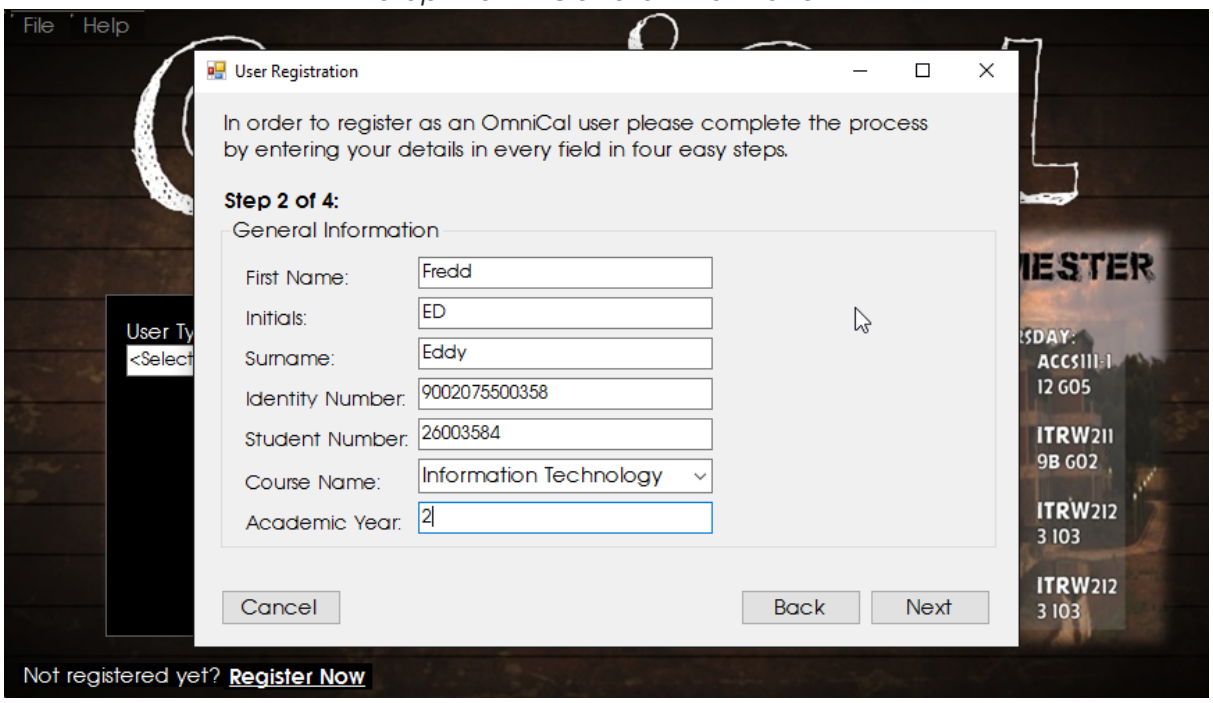
Register Screen: Allows for user input registration information

Step 1 of 4: User information



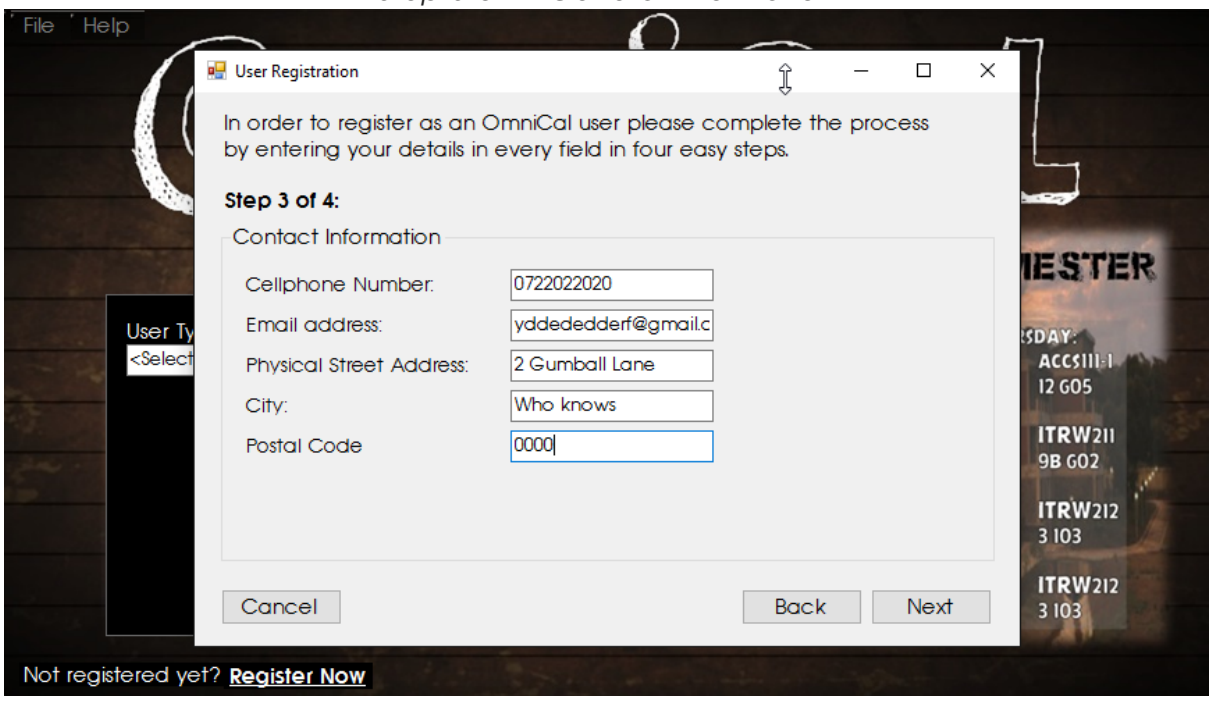
By: J Muller

Step 2 of 4: General Information



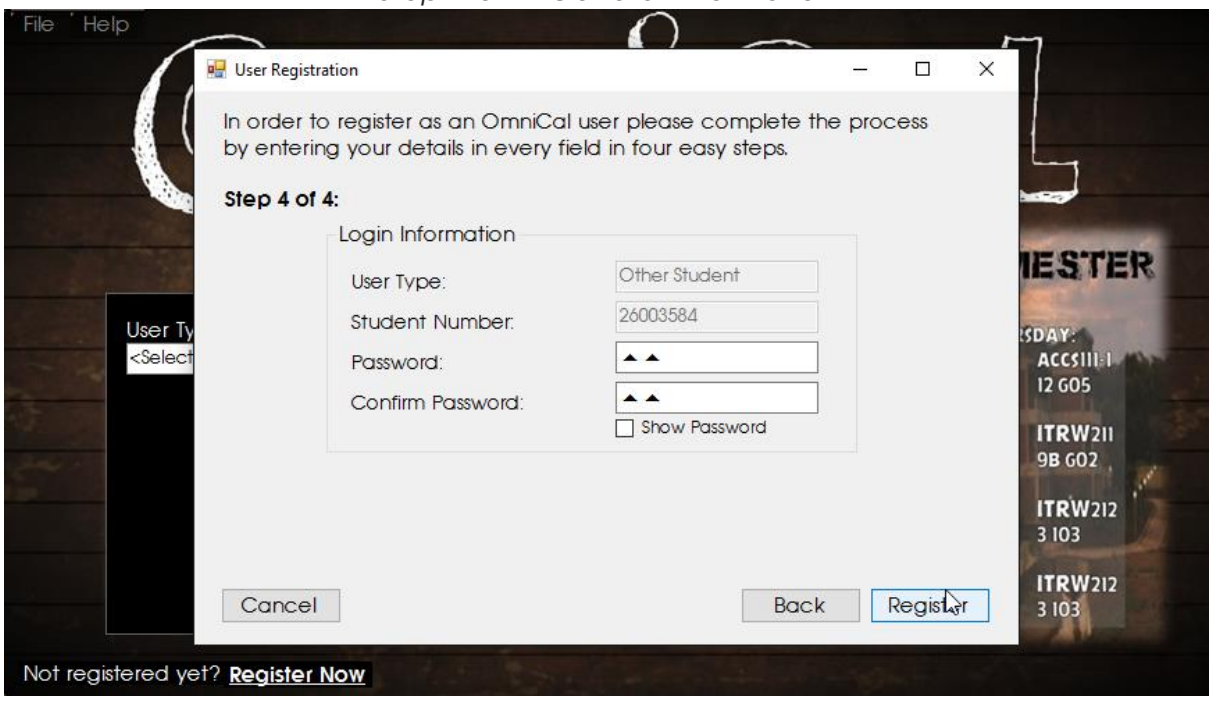
By: J Muller

Step 3 of 4: General Information



By: J Muller

Step 4 of 4: General Information



By: J Muller

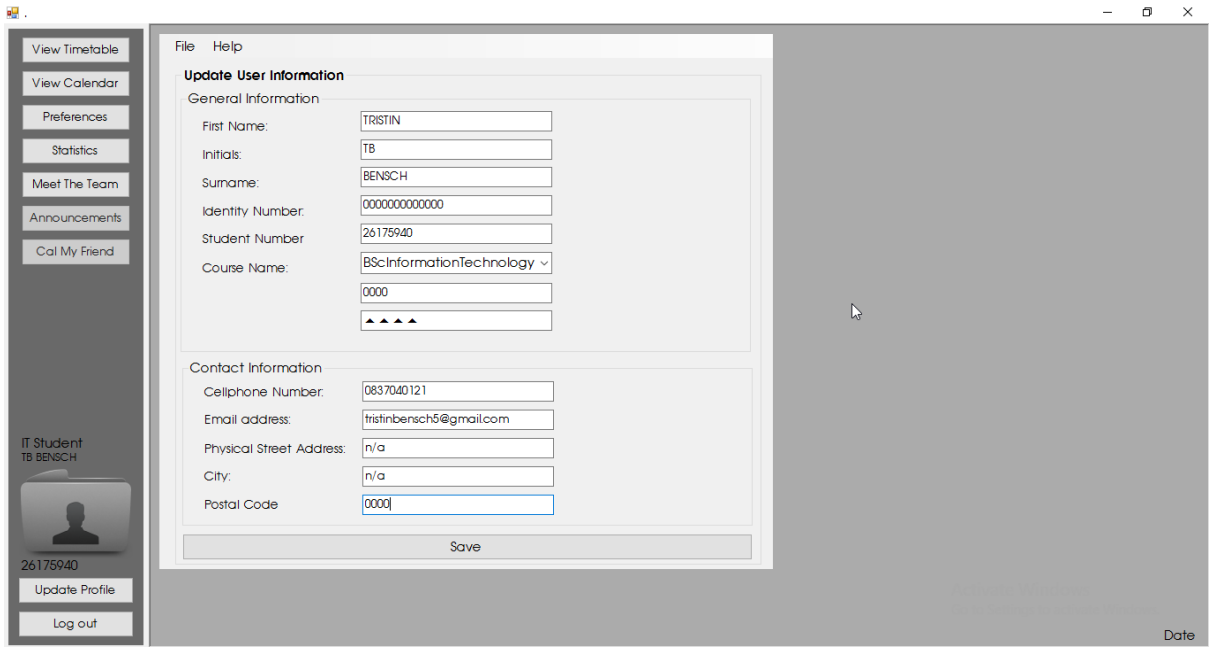
User license agreement

By: J Muller

Home Screen: Allows for user input to further navigation of OmniCal

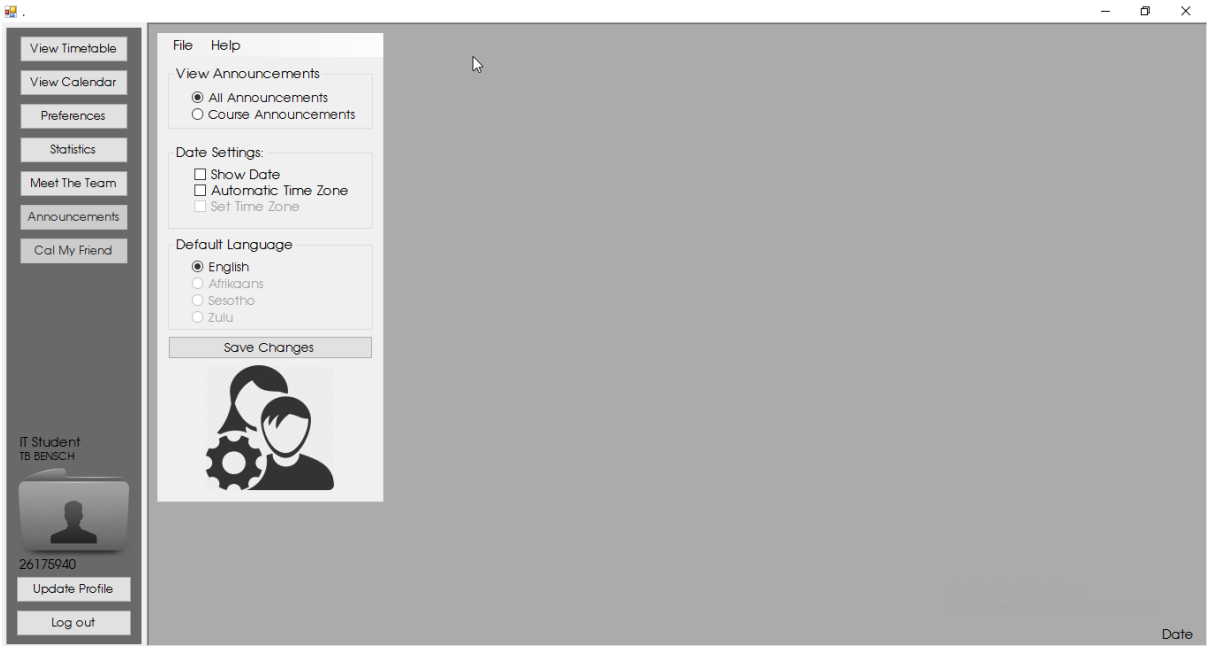
By: J Muller

Update Information Screen: Allows user input to changes details



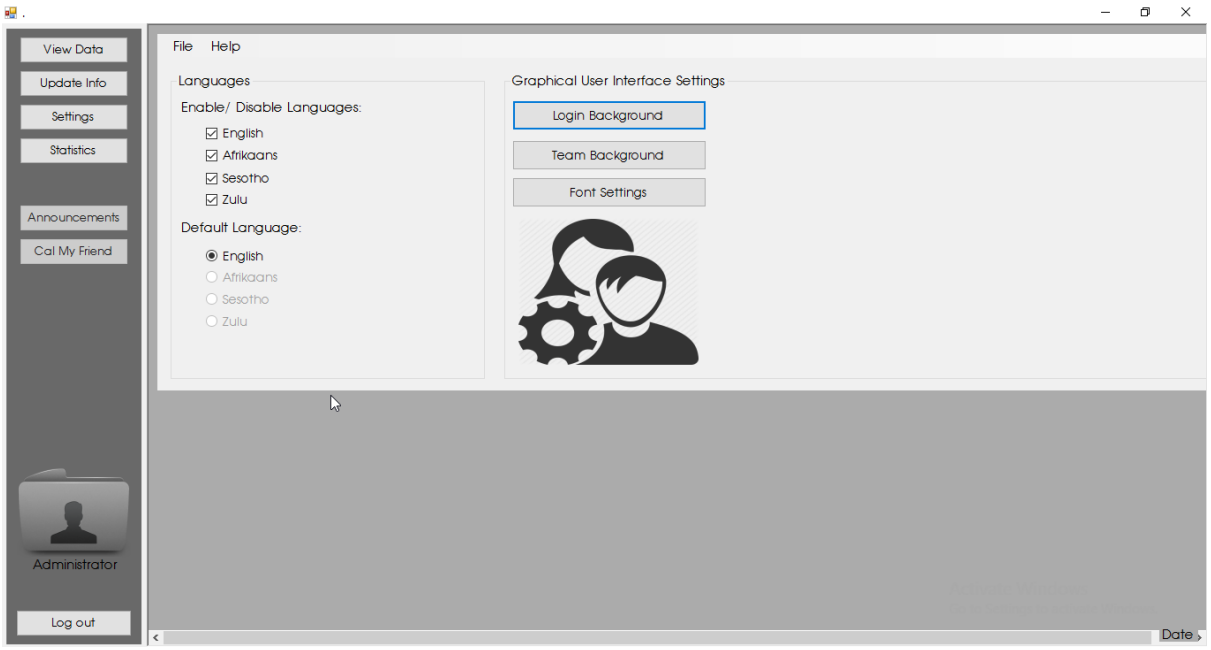
By: J Muller

User Preferences Screen: Allows user preferences input



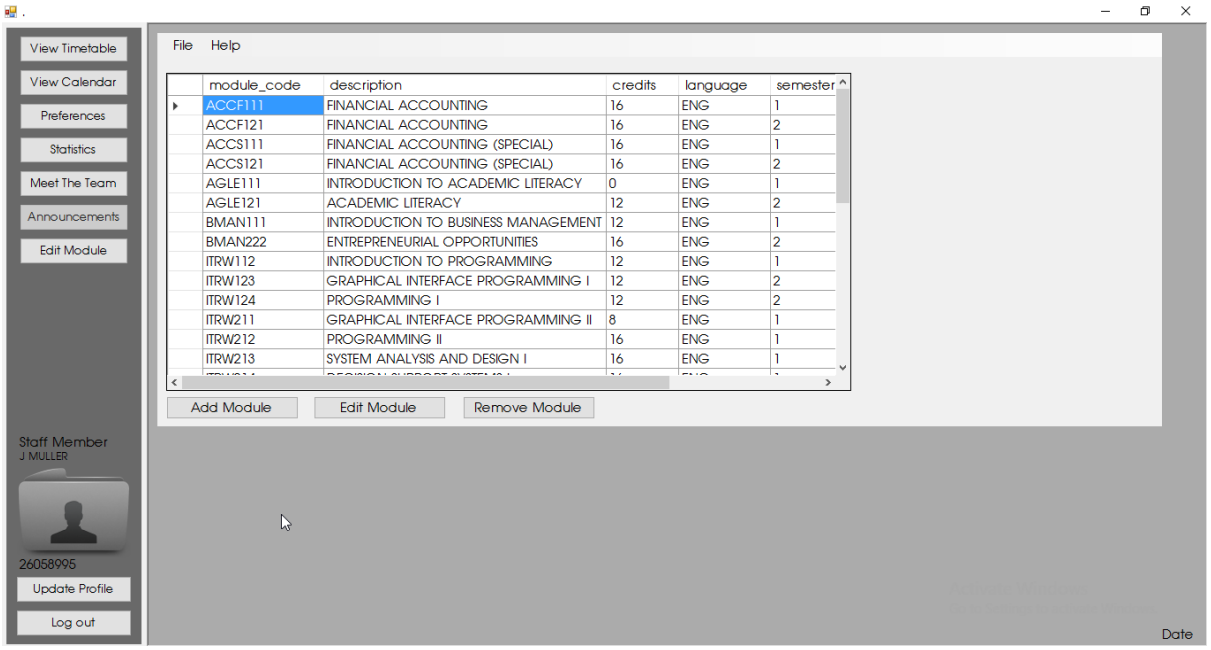
By: J Muller

Settings Screen: Allows admin to change and input system settings



By: J Muller

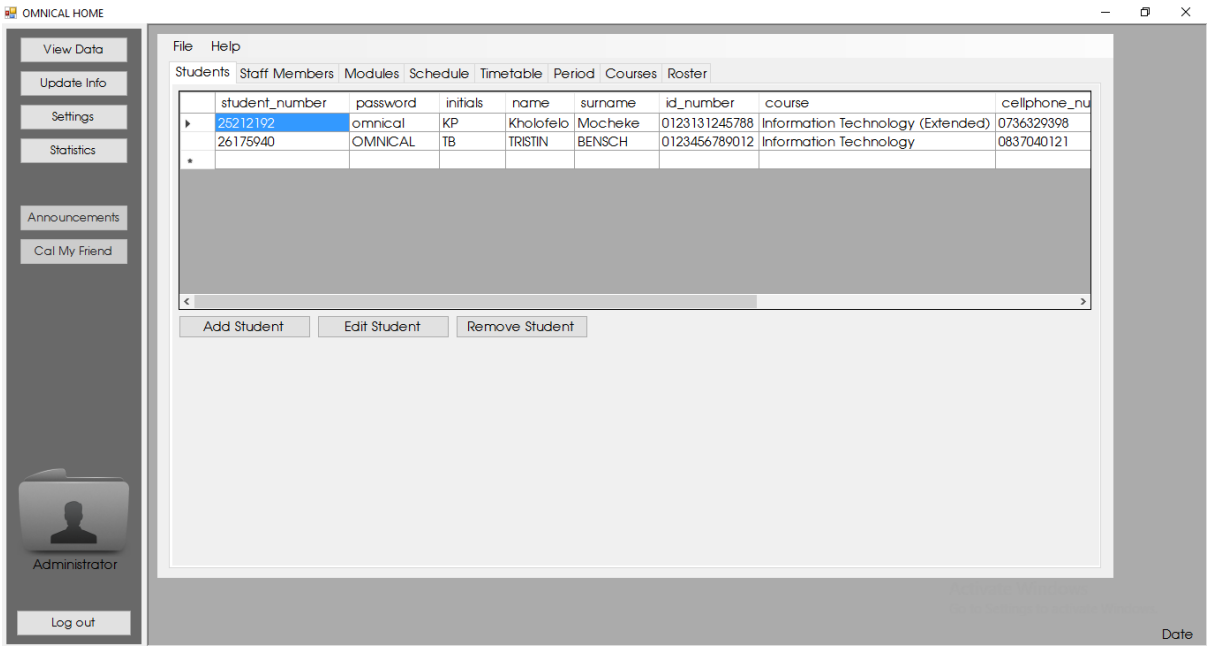
Edit modules Screen: Allows input to modules by staff



By: J Muller

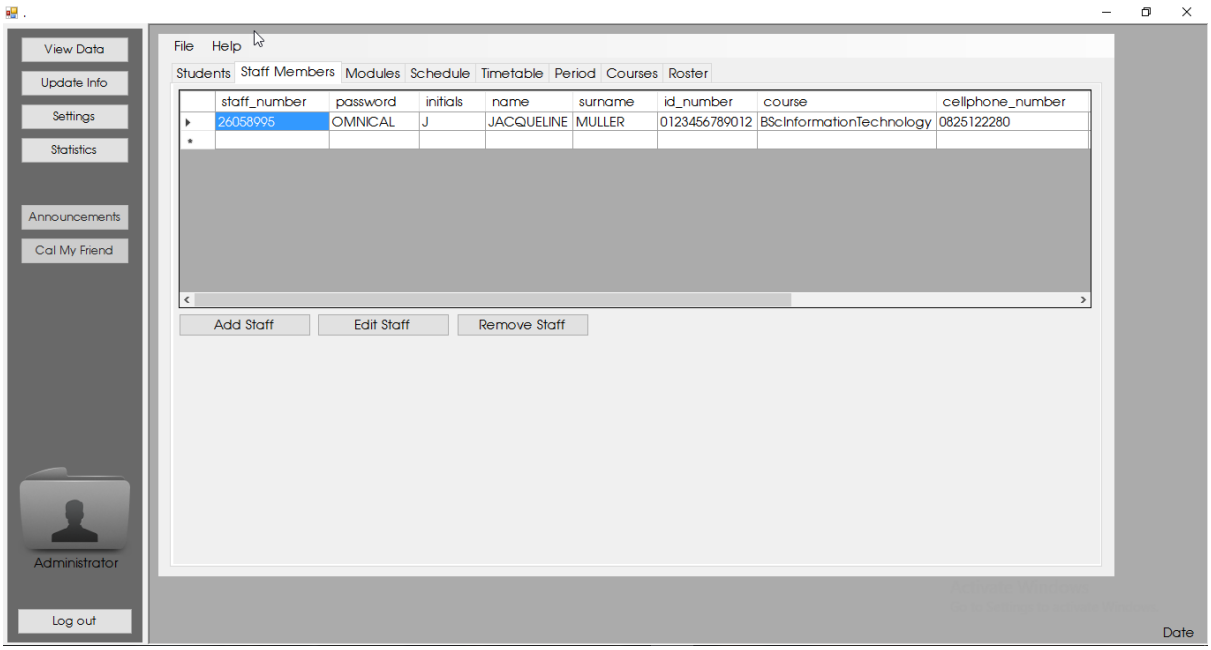
View Data Screen: Allows input for all OmniCal data by admin

Students Tab



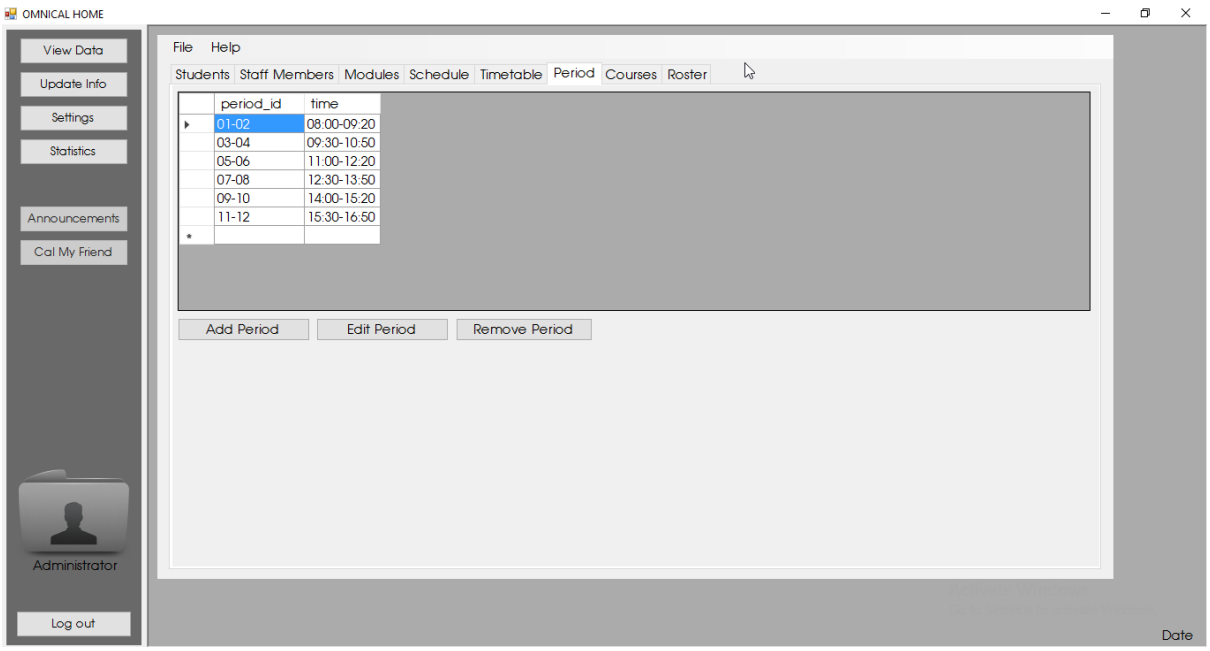
By: J Muller

Staff Tab



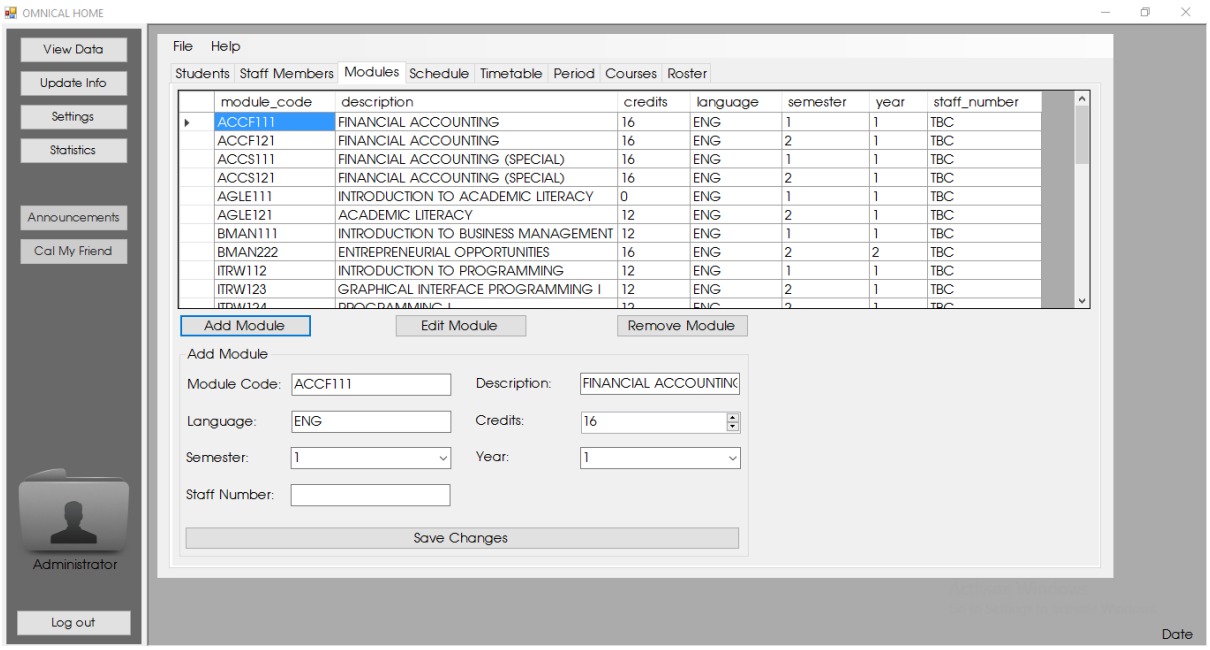
By: J Muller

Period Tab



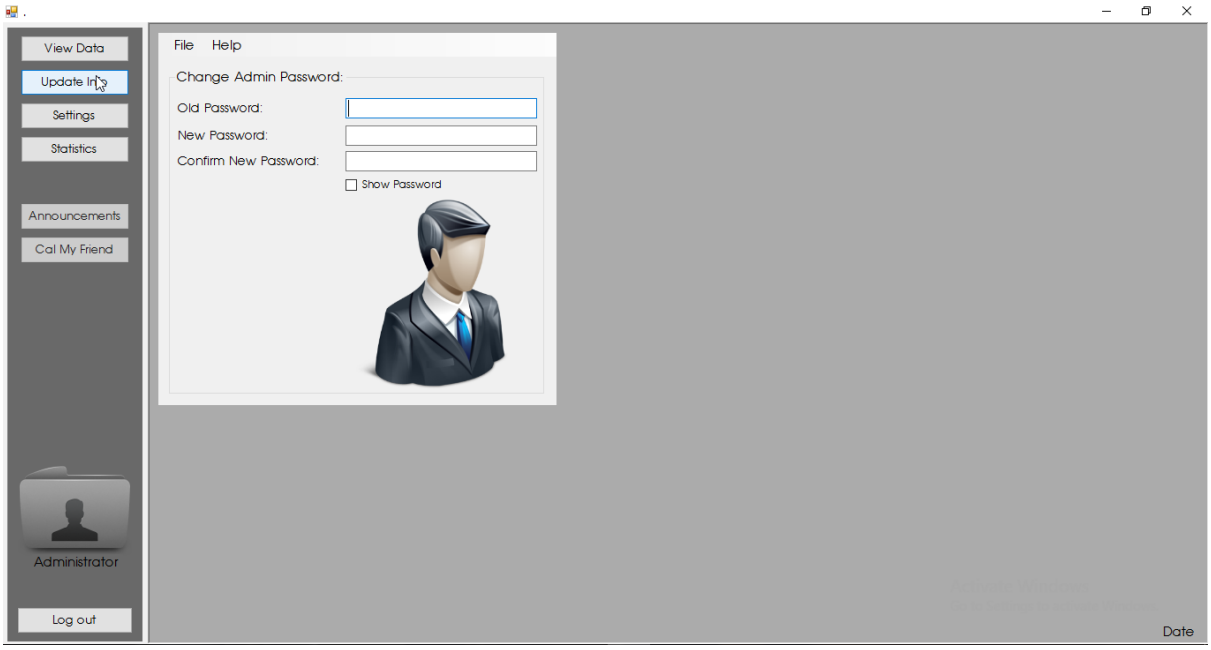
By: J Muller

Module Tab



By: J Muller

Update Admin Info Screen: Allows input OmniCal admin password settings



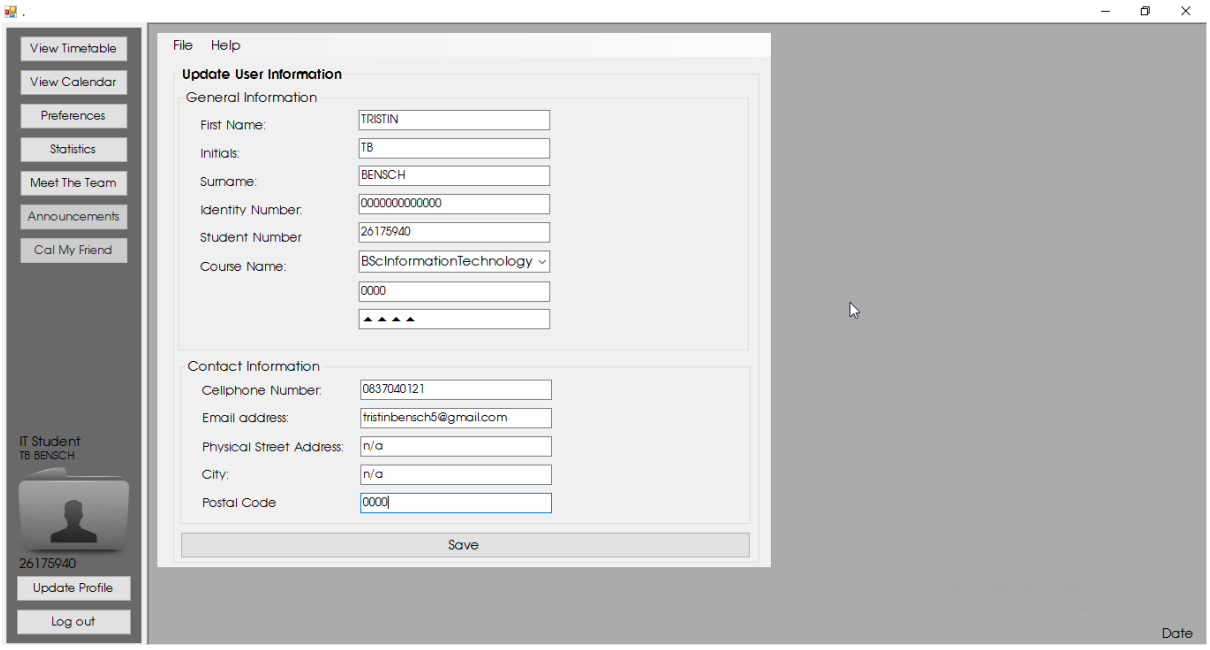
By: J Muller

Revised System Output screens

Home Screen: Outputs User details and options accordingly

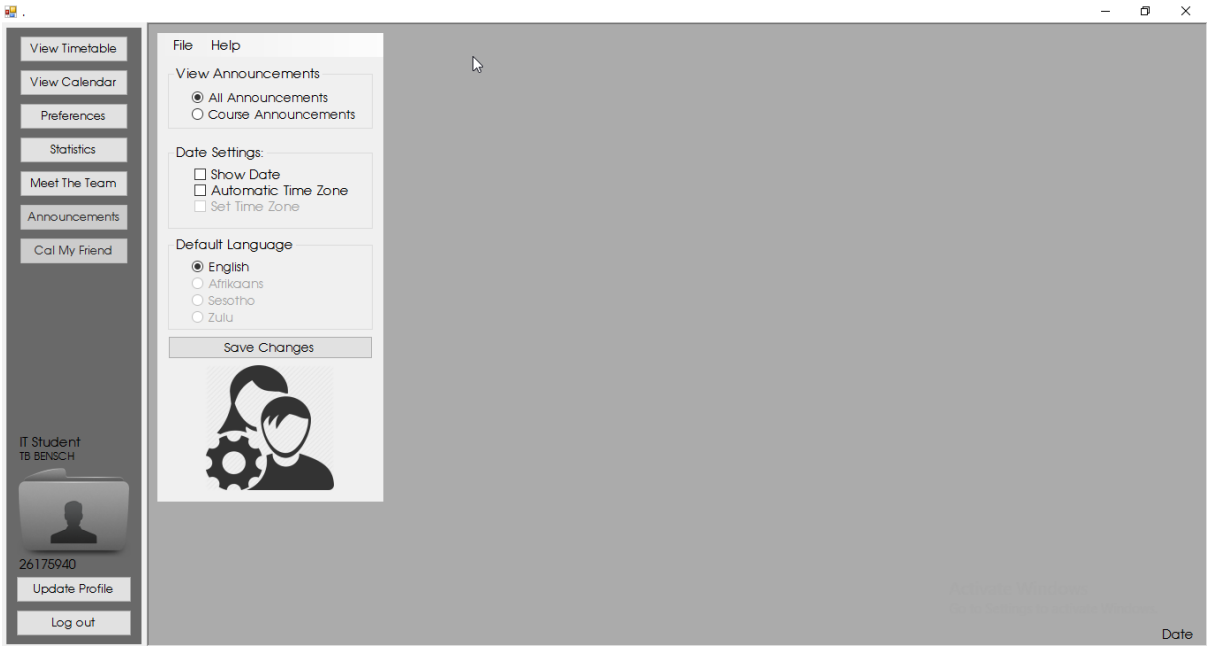
By: J Muller

Update Information Screen: Outputs User details and allows input



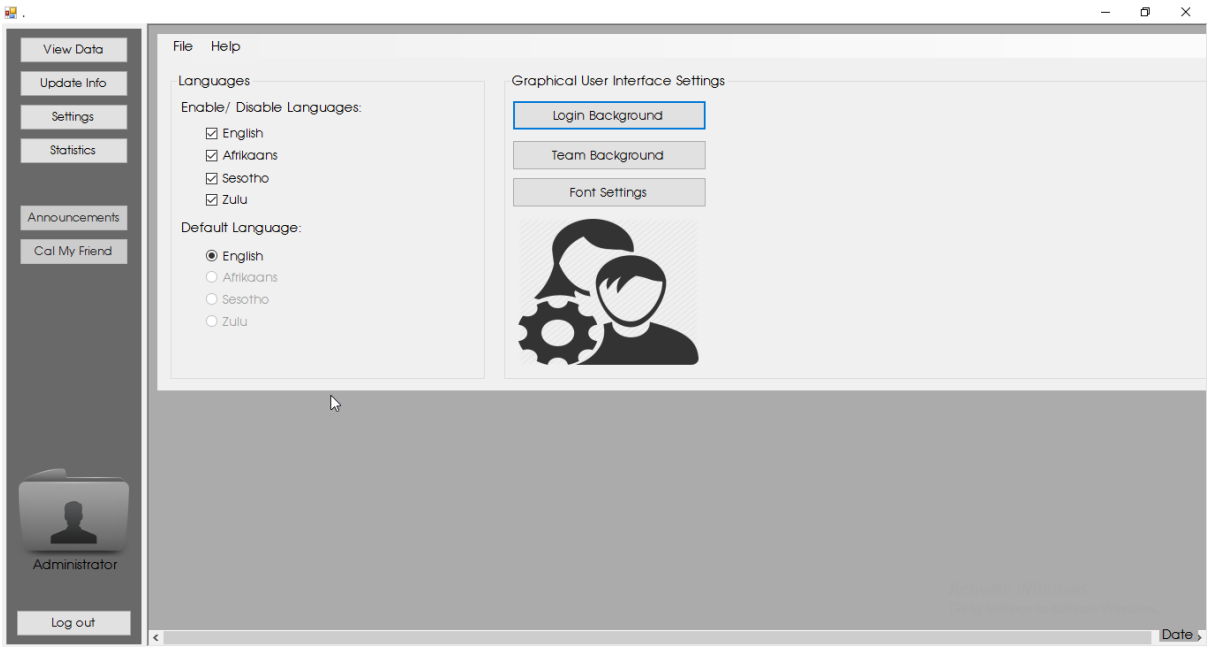
By: J Muller

User Preferences Screen: Outputs User preferences and allows input



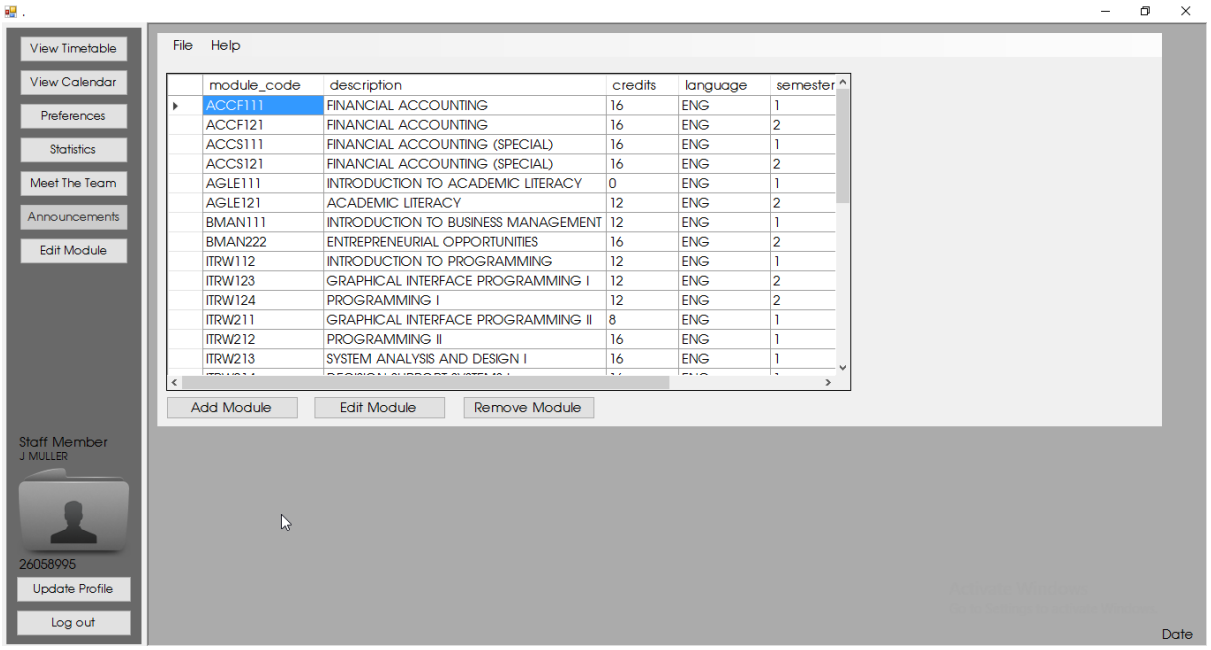
By: J Muller

Settings Screen: Outputs system settings and allows changes by admin



By: J Muller

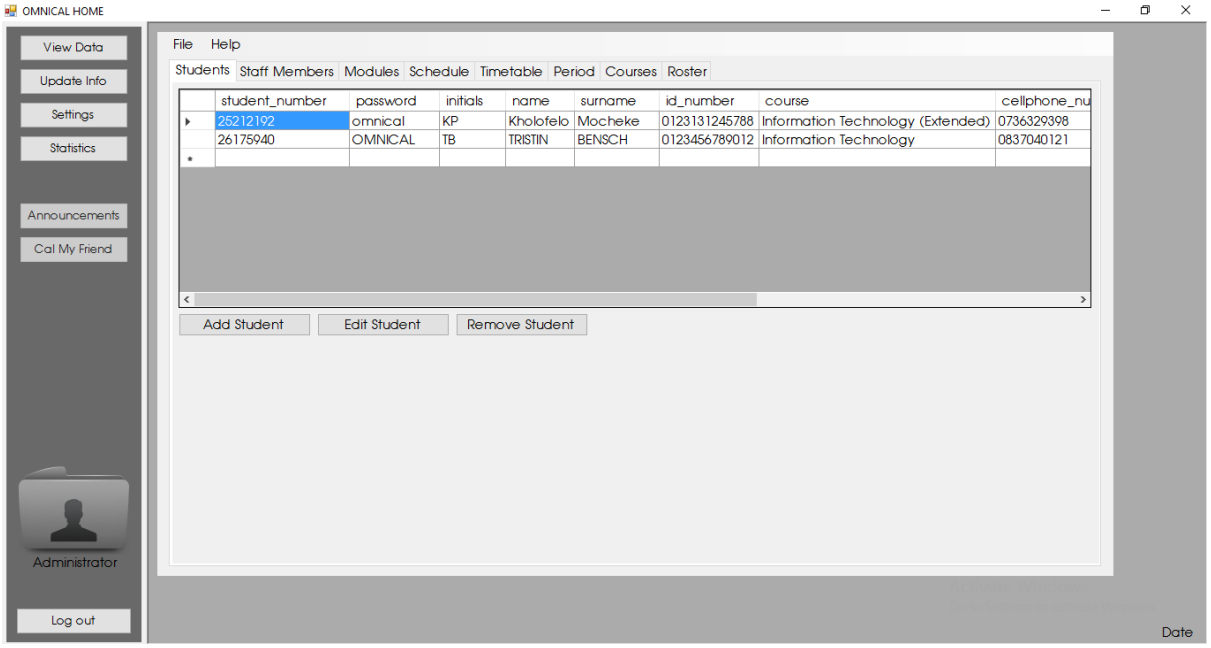
Edit modules Screen: Outputs modules and allows changes by staff



By: J Muller

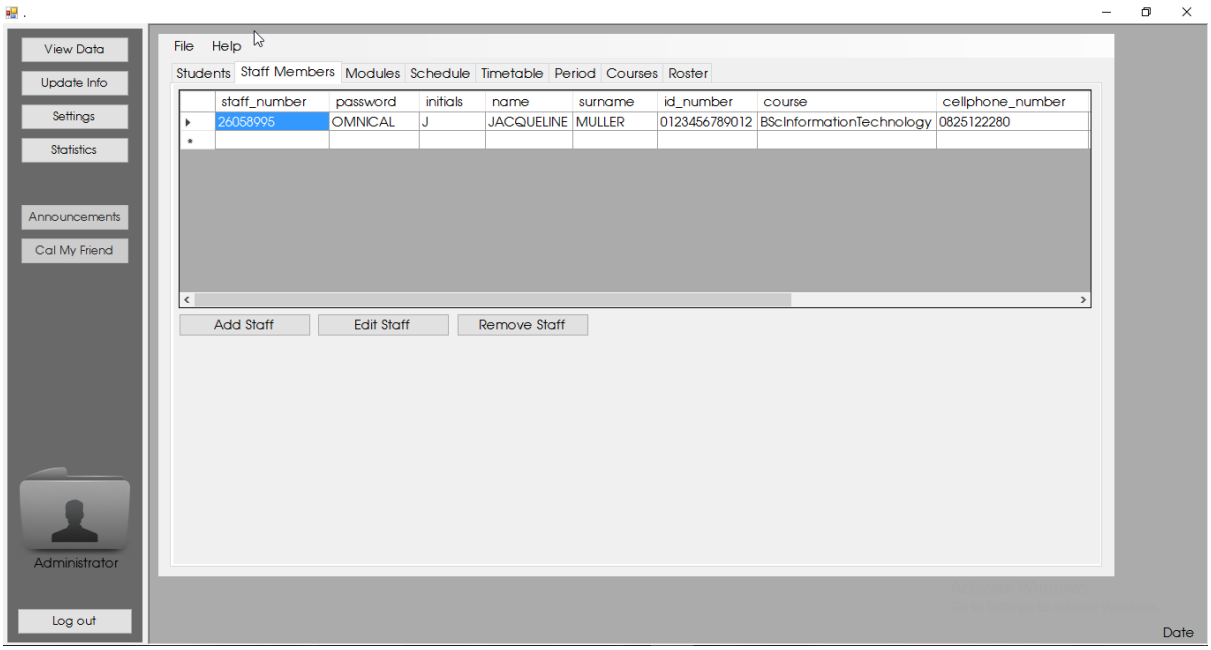
View Data Screen: Outputs all OmniCal data and allows changes by admin

Students Tab



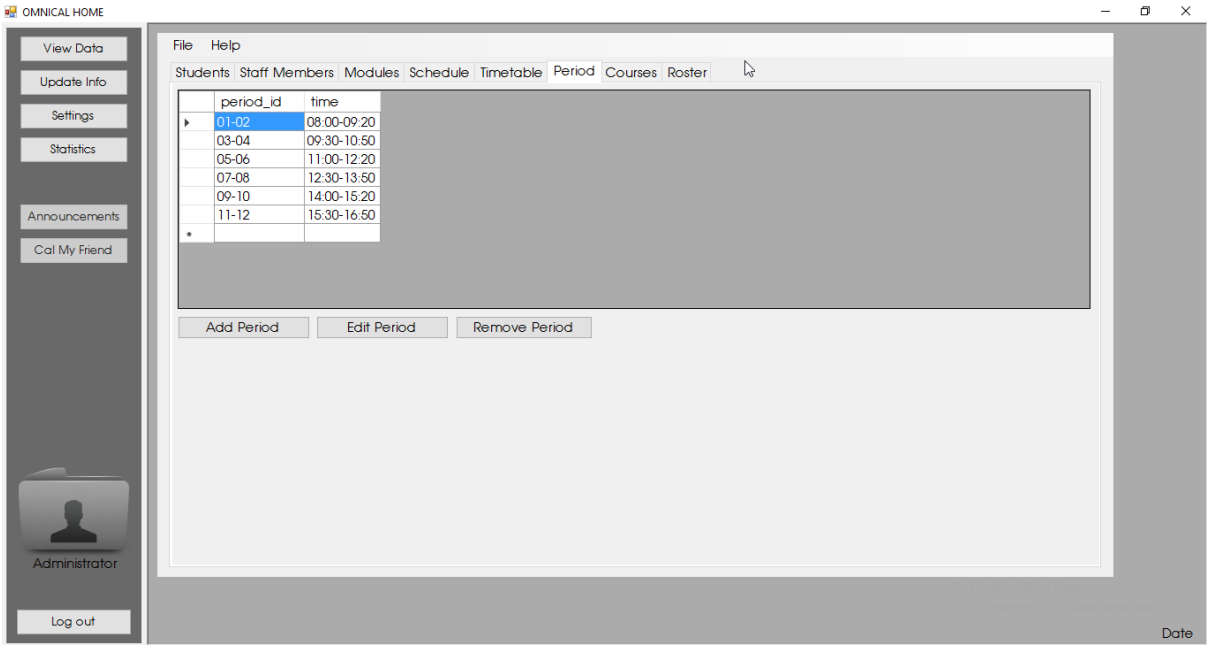
By: J Muller

Staff Tab



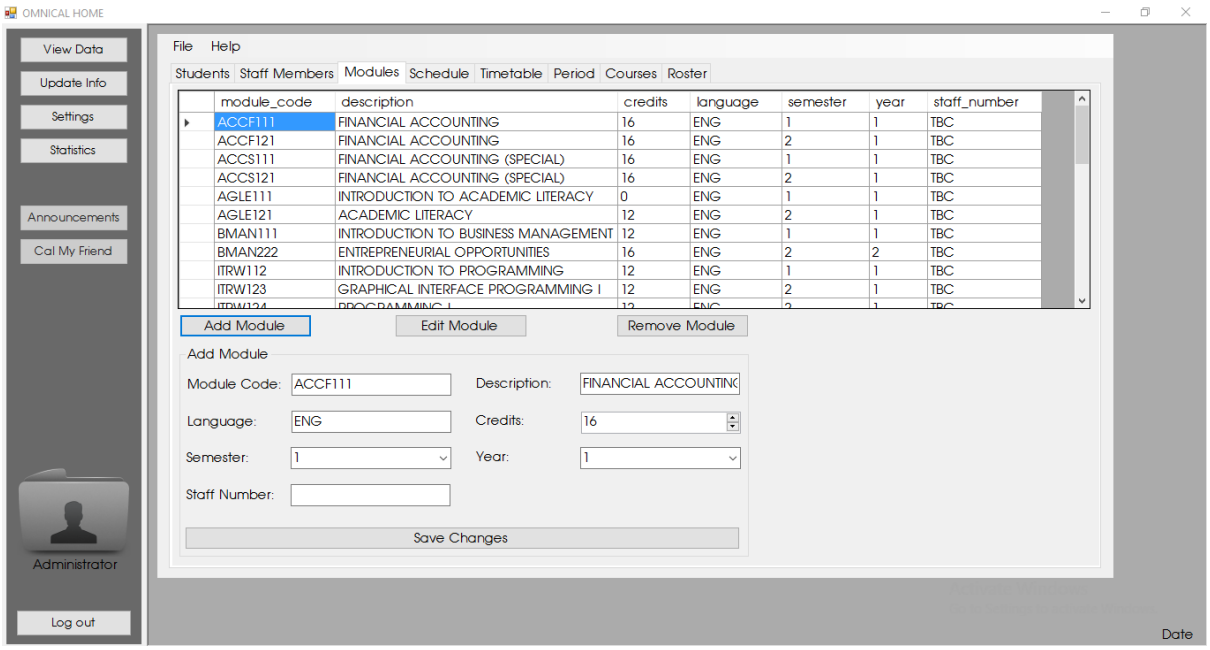
By: J Muller

Period Tab



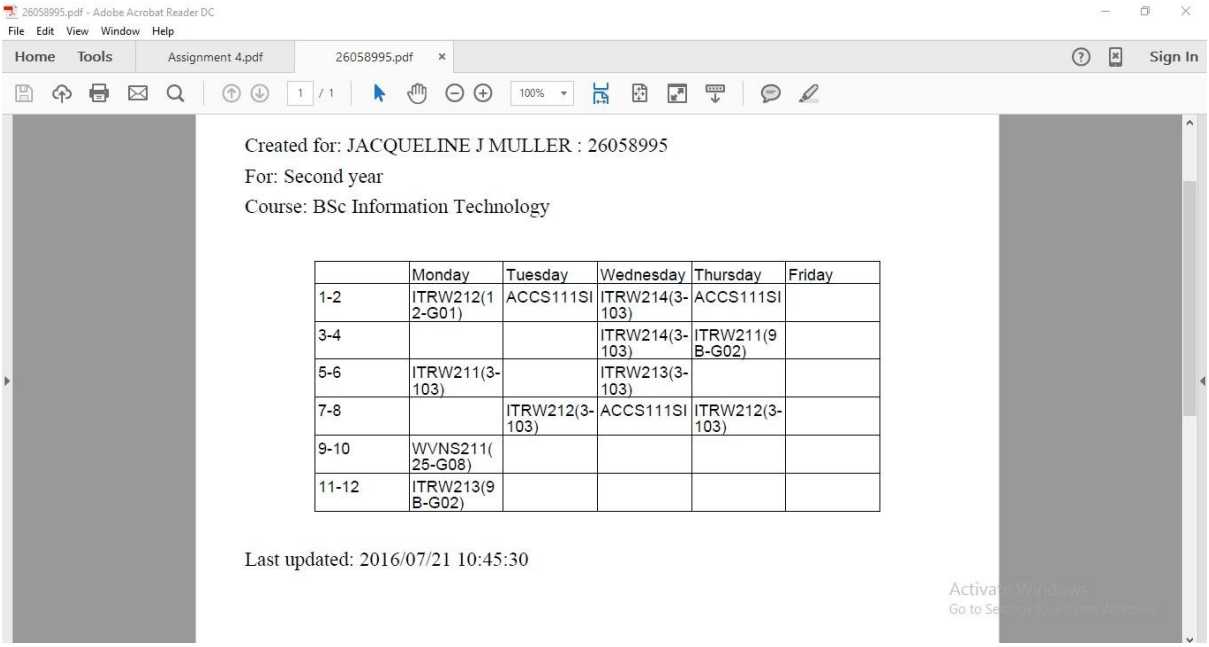
By: J Muller

Module Tab



By: J Muller

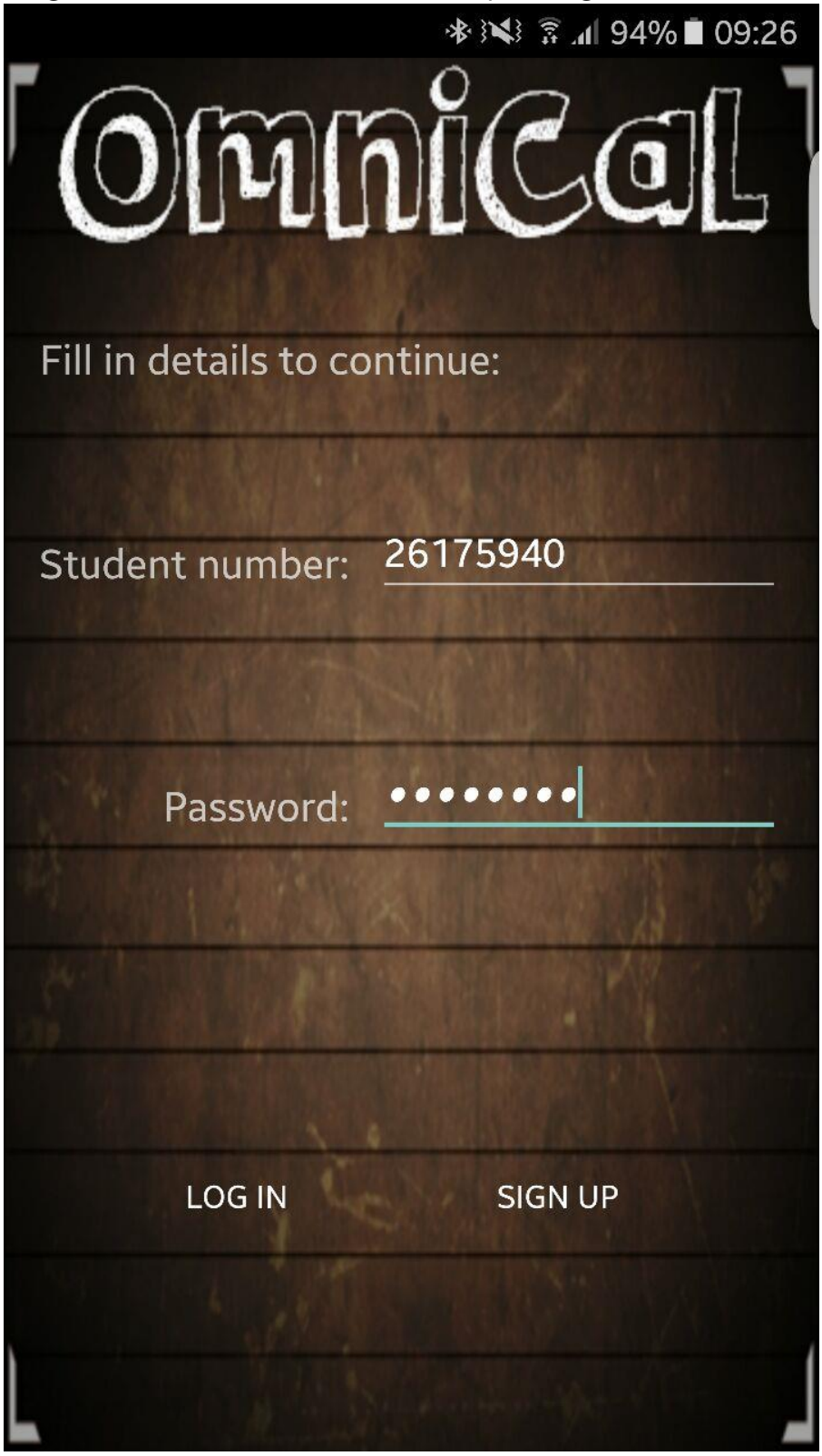
PDF: Outputs User timetable in PDF format



By: J Muller

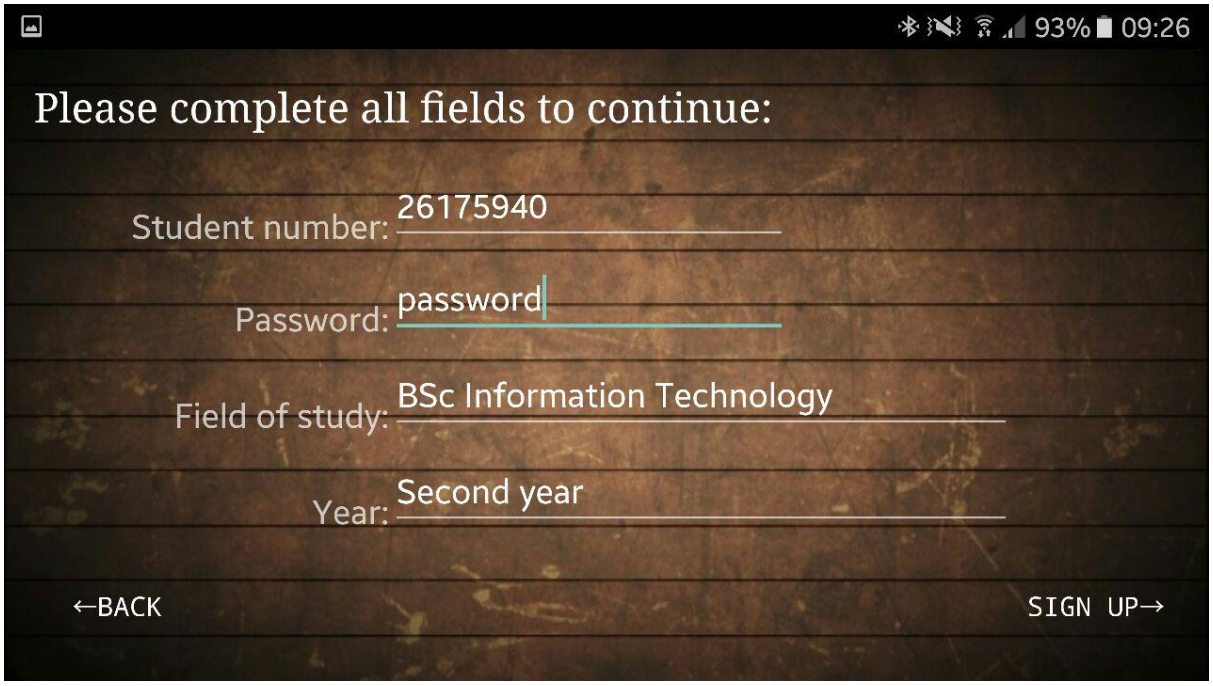
Revised App Input screens

Log in Screen: Allows for user input log in information



By: TB Bensch

Registration Screen: Allows for user input for registration purposes



By: TB Bensch

Revised System Menus

Login Screen: Main Menu

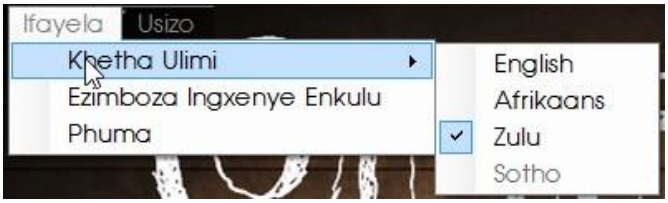
File (English)

By: J Muller

File (Afrikaans)

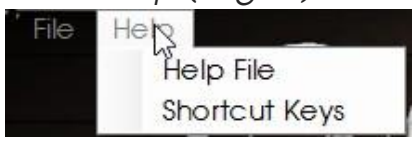
By: J Muller

File (Zulu)



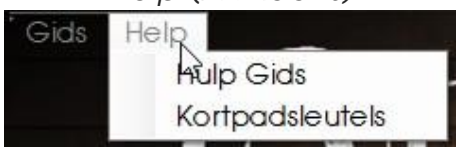
By: J Muller

Help (English)



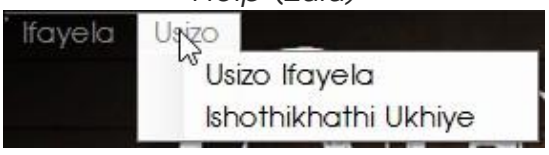
By: J Muller

Help (Afrikaans)



By: J Muller

Help (Zulu)



By: J Muller

Login Screen: Social Media Menu

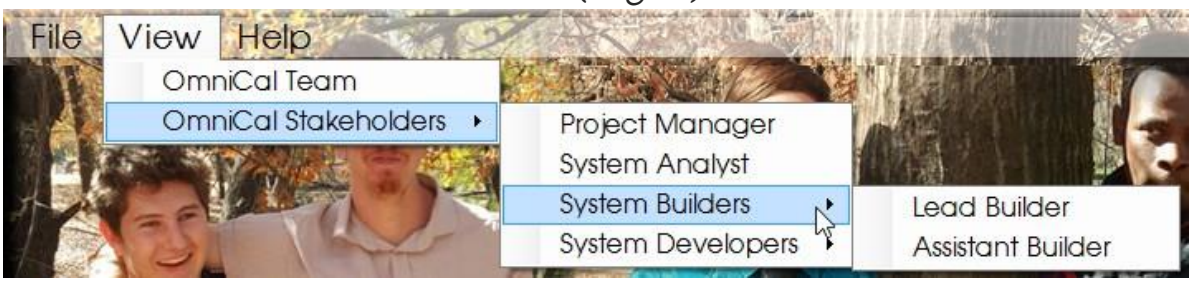
By: J Muller

Meet The Team Screen: Main Menu

File (English)

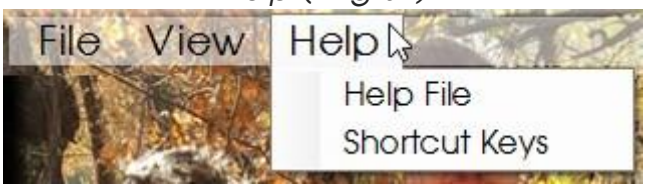
By: J Muller

View (English)



By: J Muller

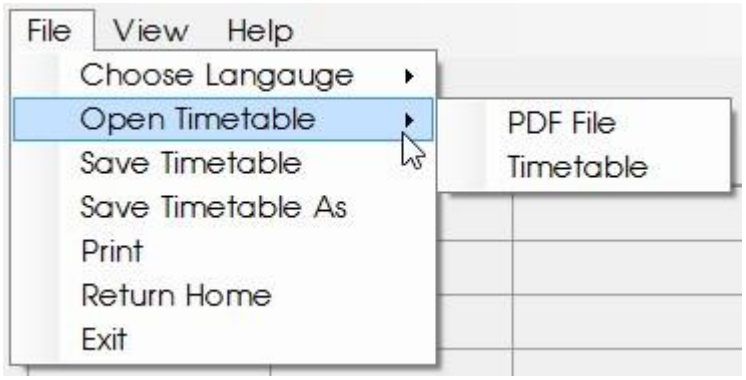
Help (English)



By: J Muller

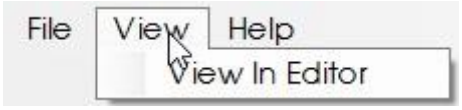
Timetable: Main Menu

File (English)



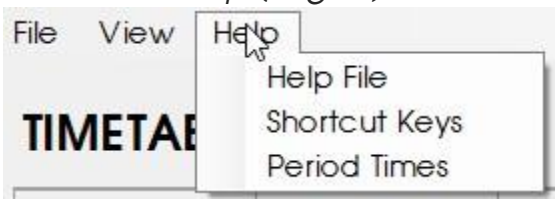
By: J Muller

View (English)



By: J Muller

Help (English)



By: J Muller

Main Menu for the rest of the system

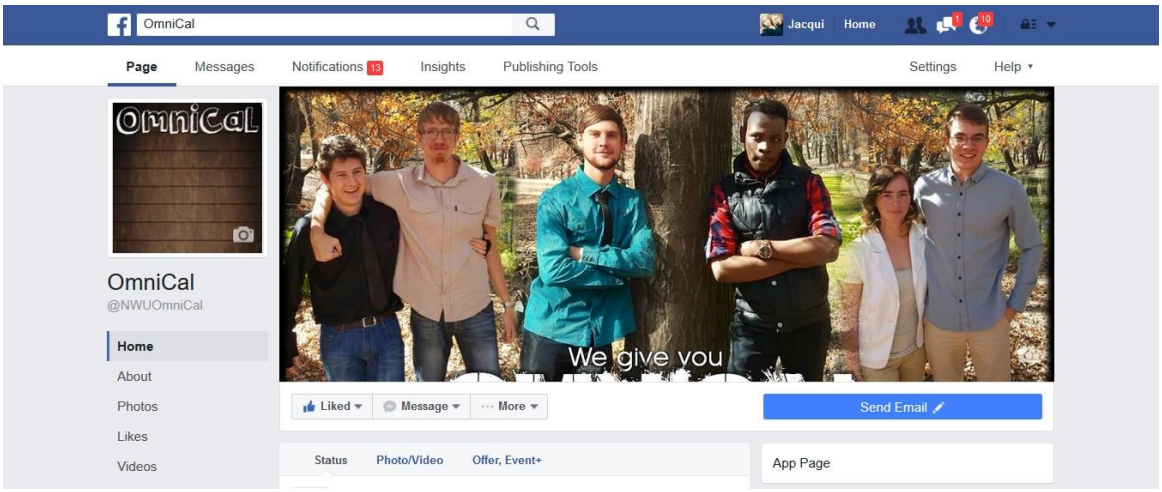
The forms that have not been mentioned in the above menu screenshots follow the exact same layout as that of the login screen main menu.

By: J Muller

Revised Desktop Online Input/ Output Screens

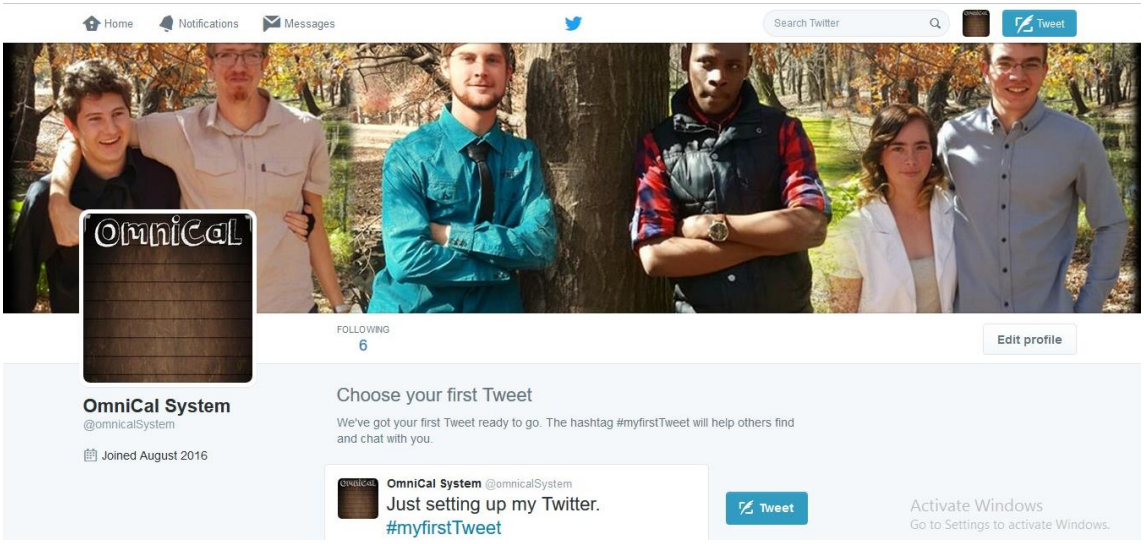
Social Media

Facebook



By: J Muller

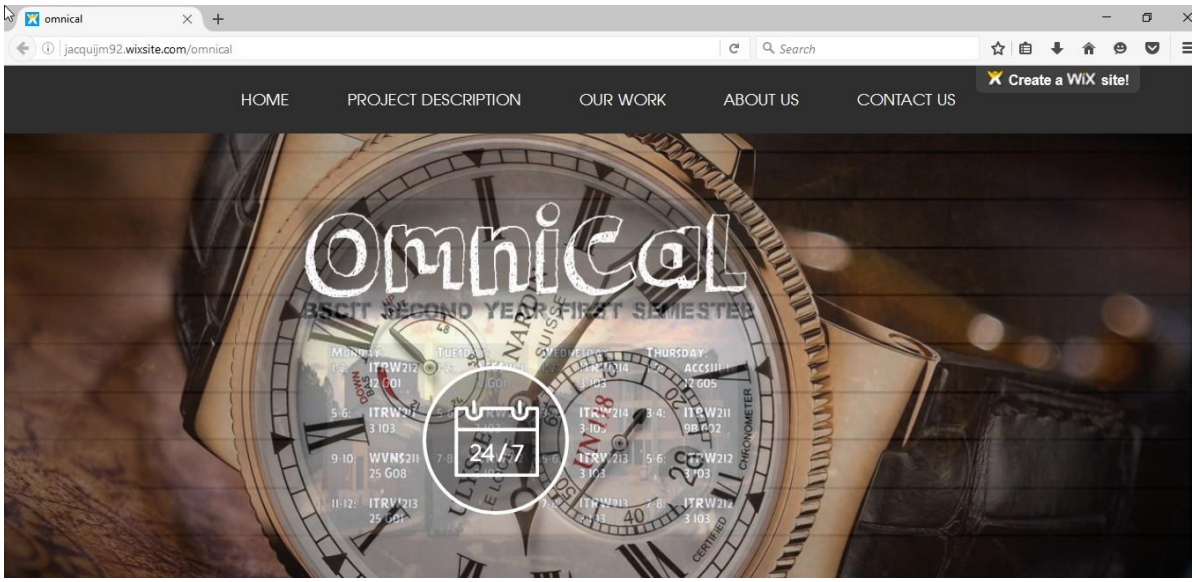
Twitter



By: J Muller

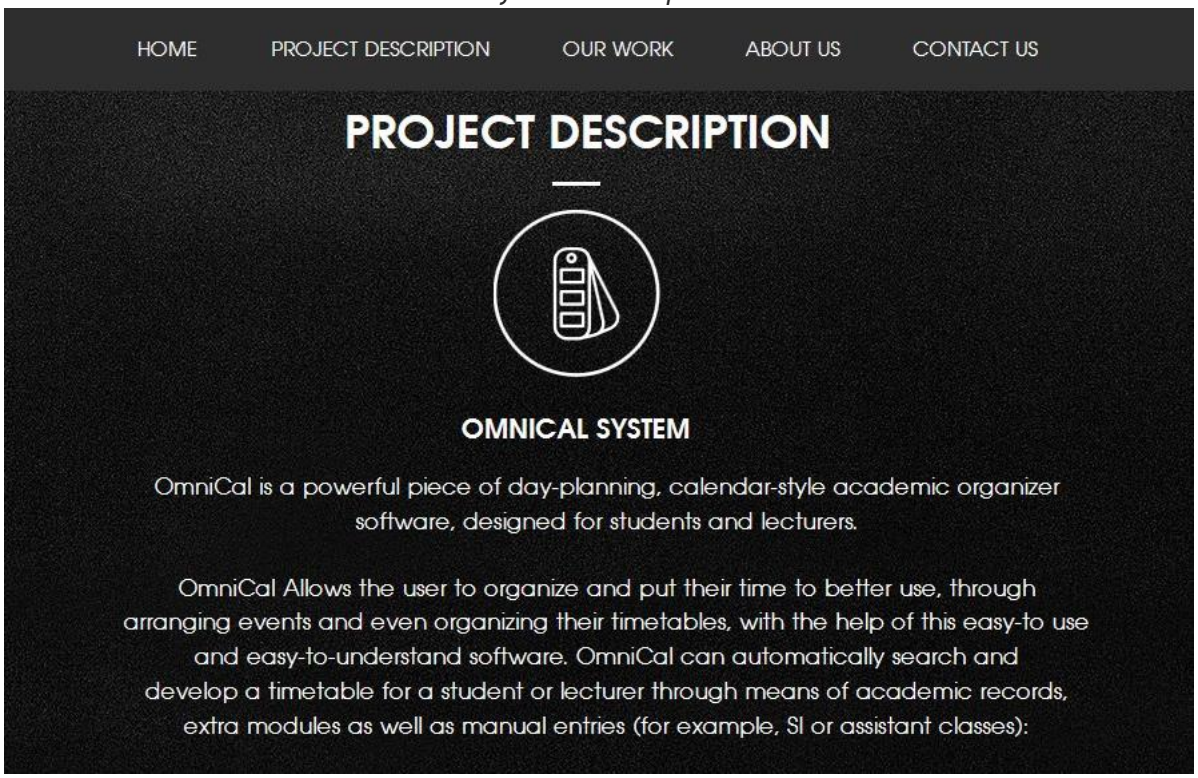
Website (Desktop View)

Home



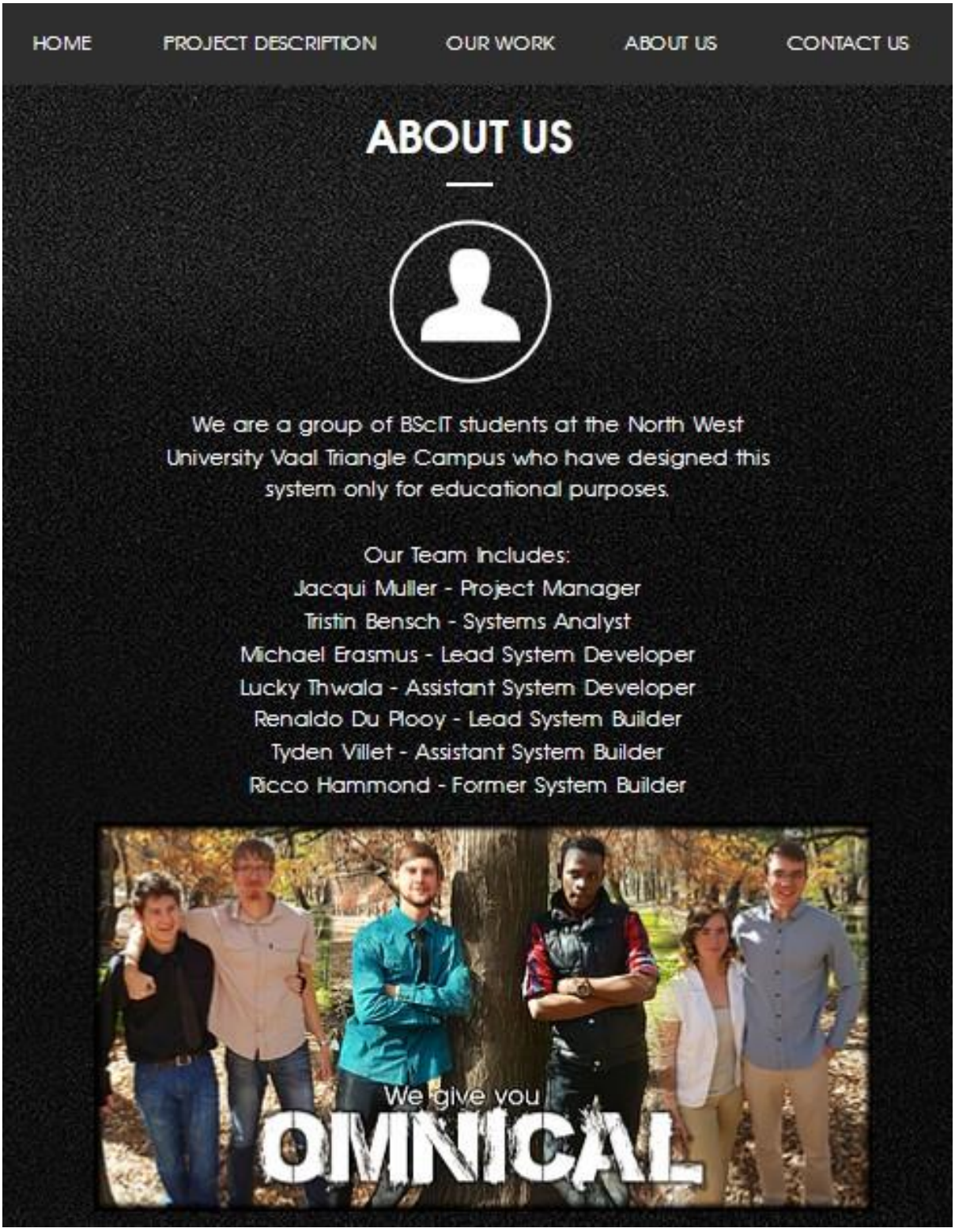
By: J Muller

Project Description



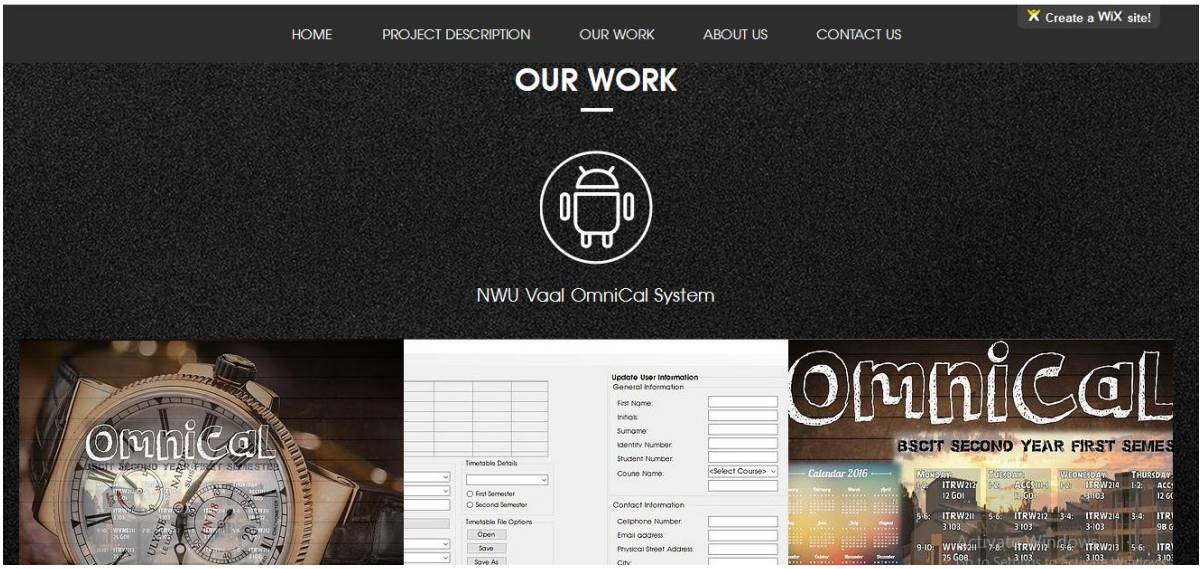
By: J Muller

About Us



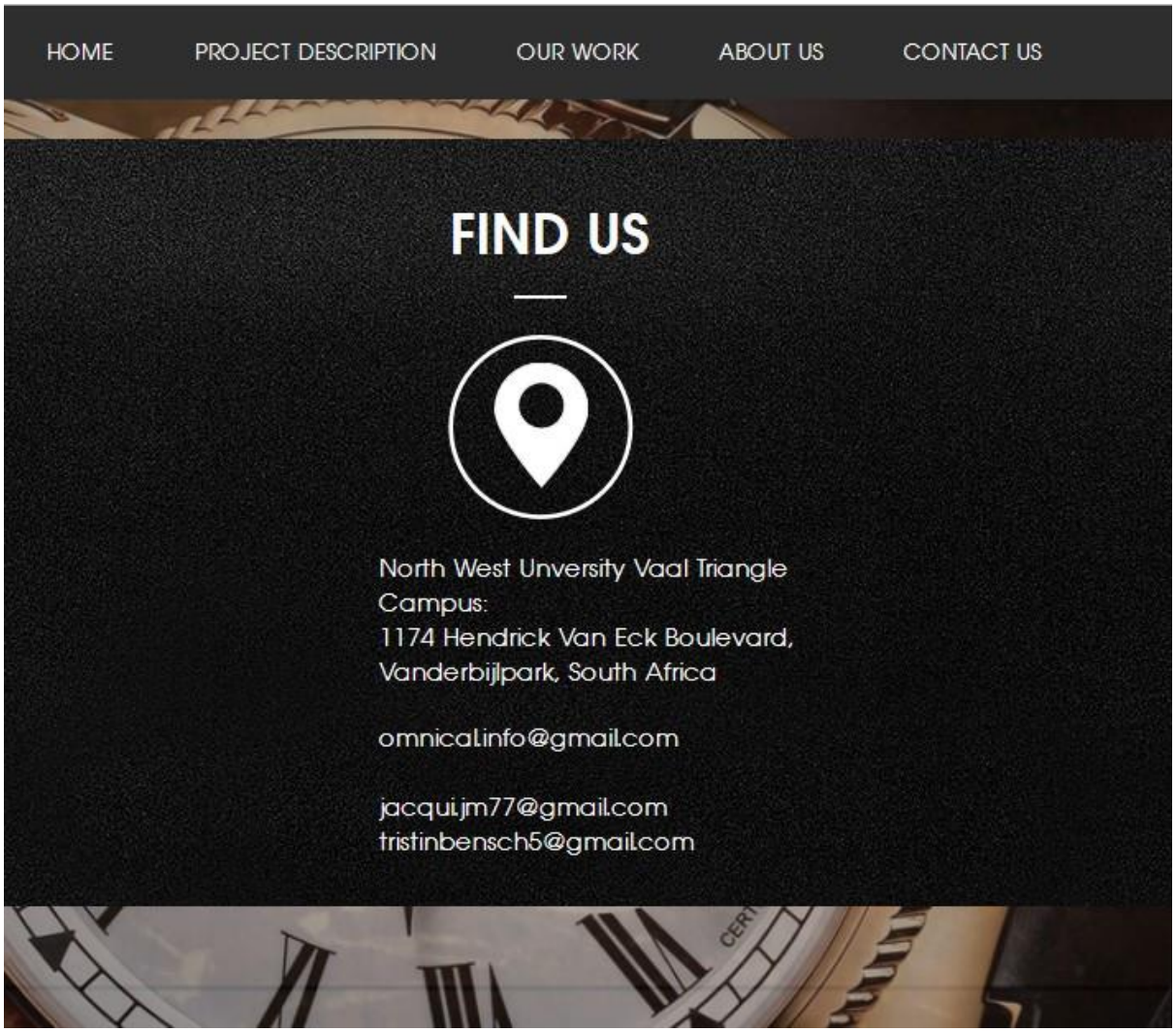
By: J Muller

Our Work



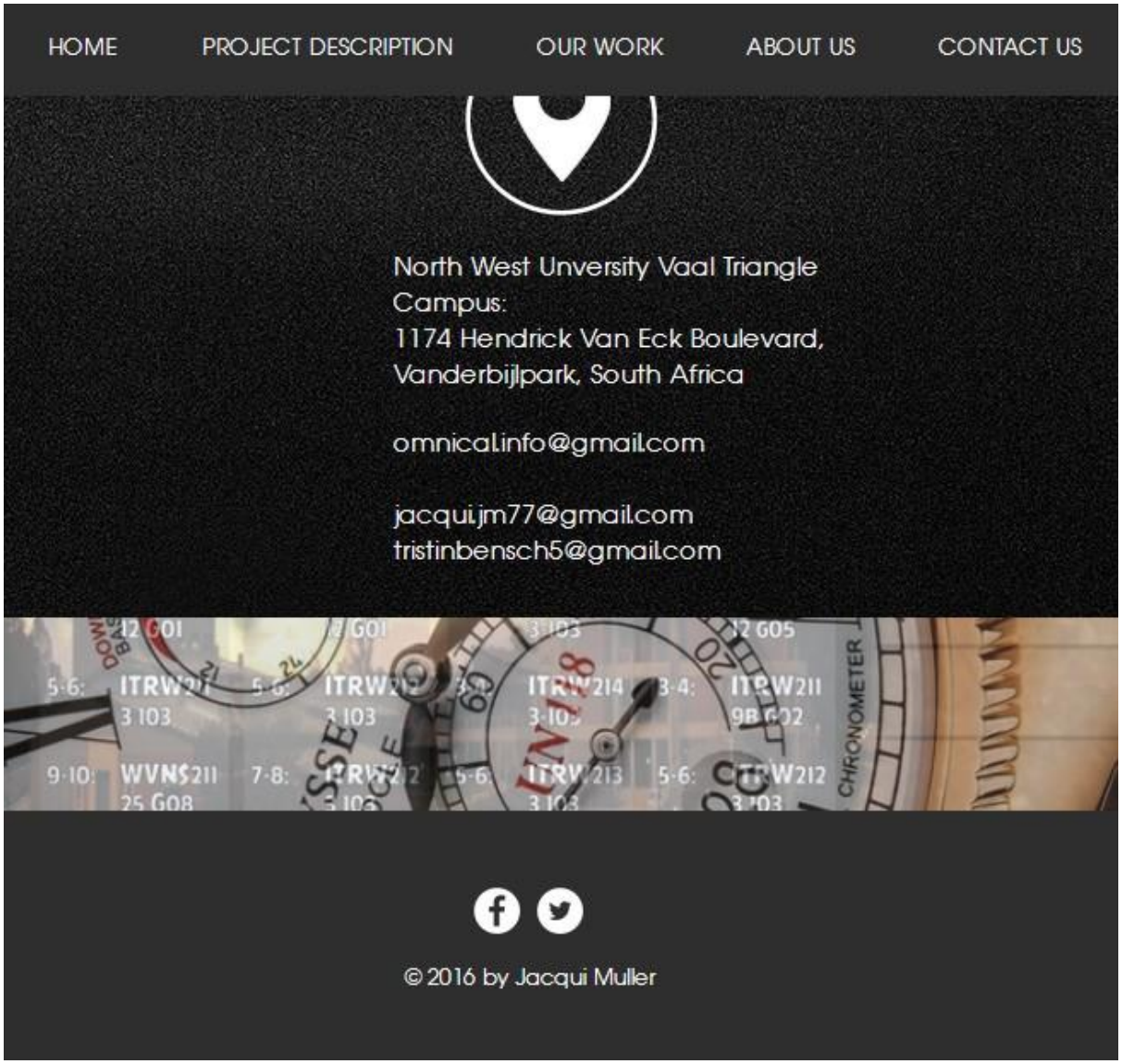
By: J Muller

Contact Us



By: J Muller

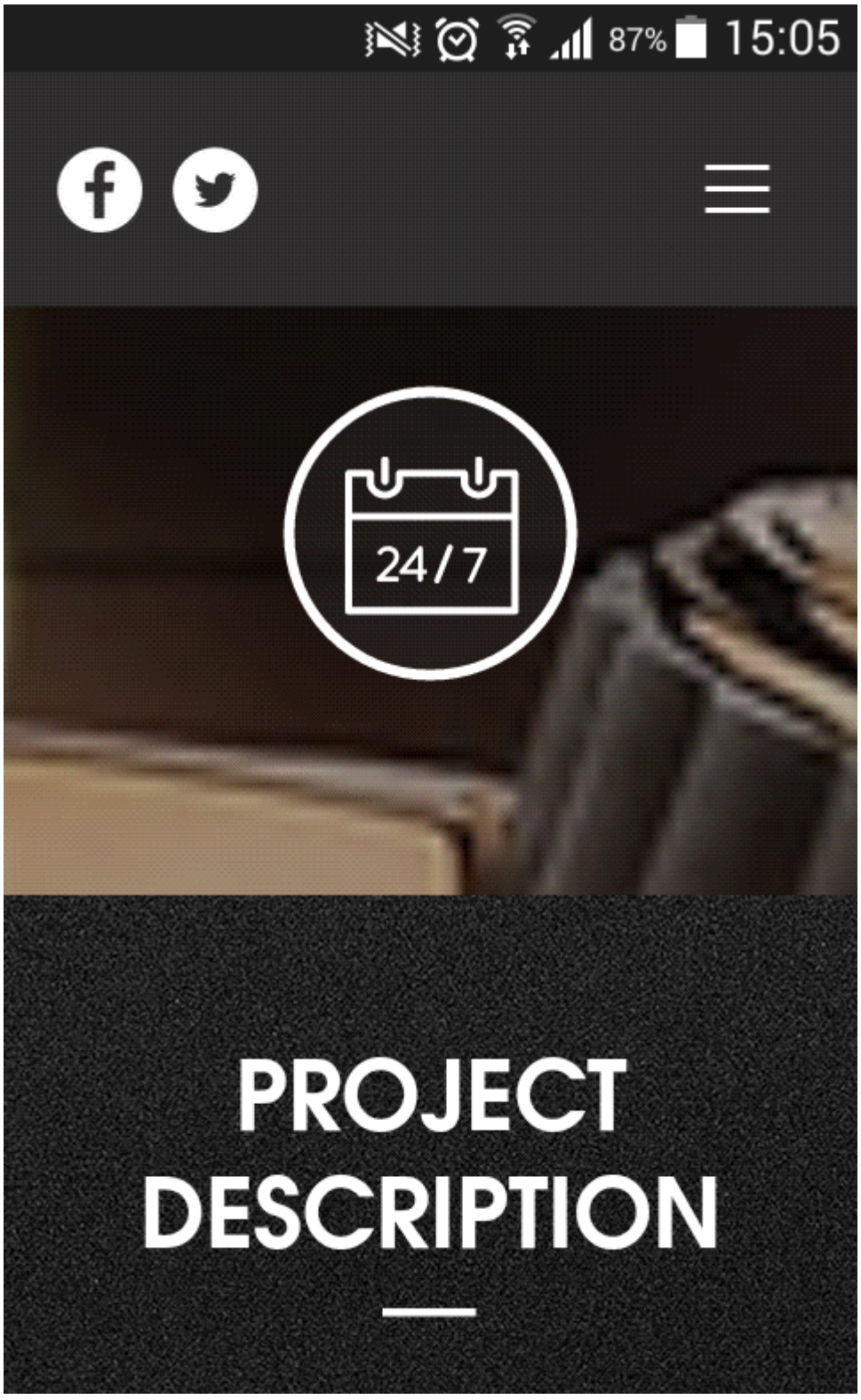
Social Media Links



By: J Muller

Website (Mobile View)

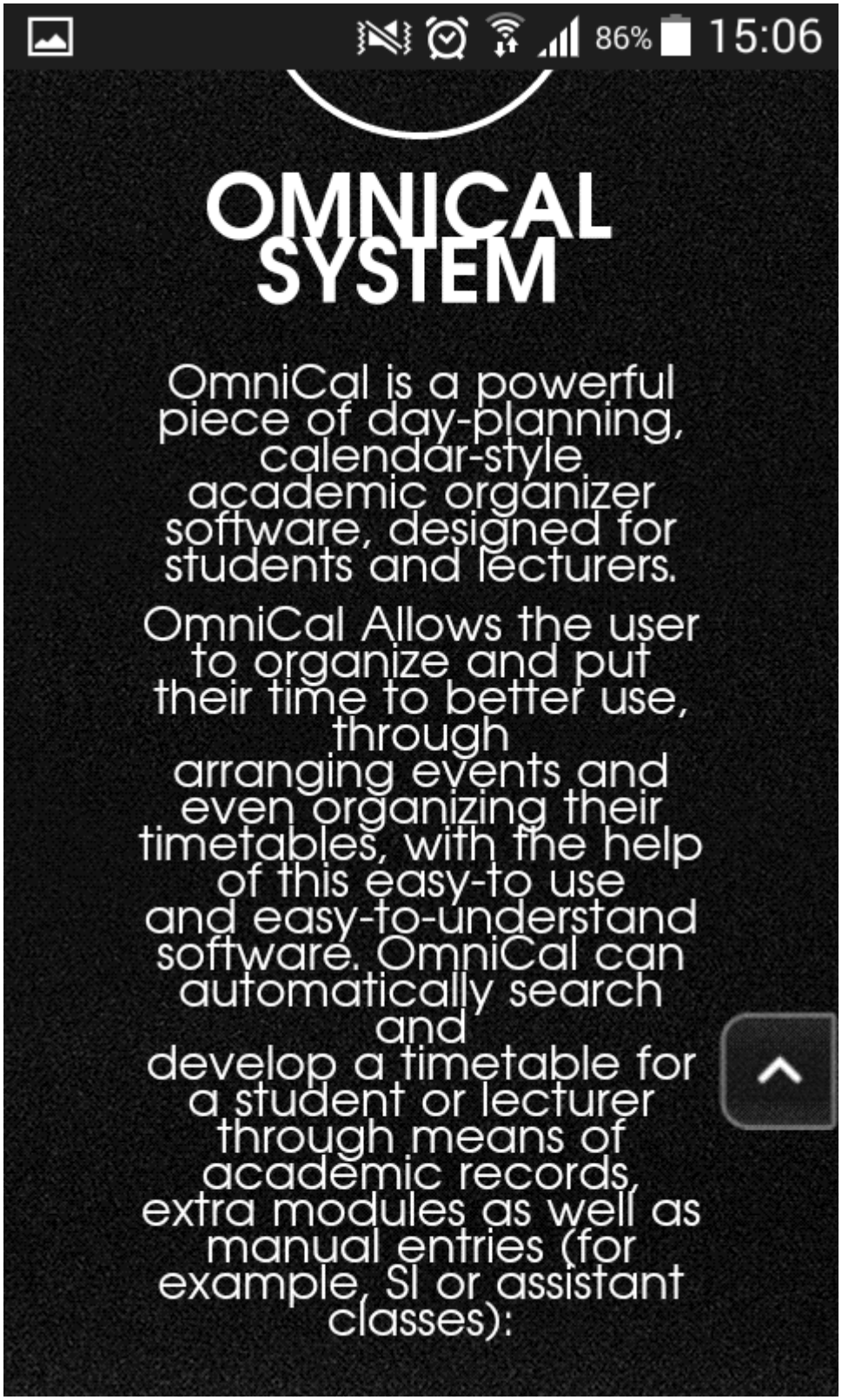
Home



By: J Muller


86%
15:06
PROJECT
DESCRIPTION
OMNICAL
SYSTEM
OmniCal is a powerful piece of day-planning, calendar-style academic organizer software, designed for students and lecturers.
OmniCal Allows the user to organize and put their time to better use,

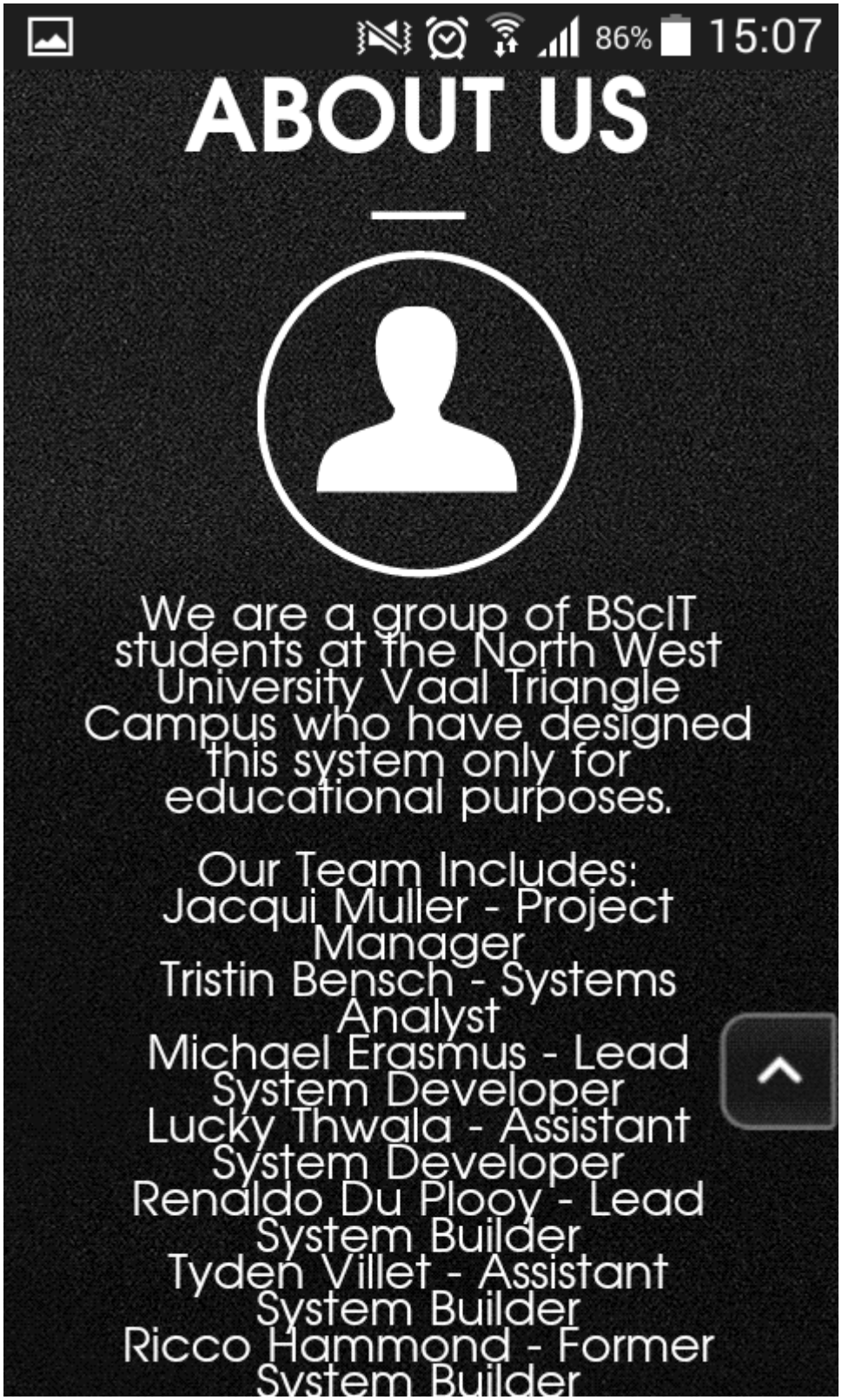
By: J Muller


86%
15:06
OMNICAL SYSTEM
OmniCal is a powerful piece of day-planning, calendar-style academic organizer software, designed for students and lecturers.
OmniCal Allows the user to organize and put their time to better use, through arranging events and even organizing their timetables, with the help of this easy-to use and easy-to-understand software. OmniCal can automatically search and develop a timetable for a student or lecturer through means of academic records, extra modules as well as manual entries (for example, SI or assistant classes):

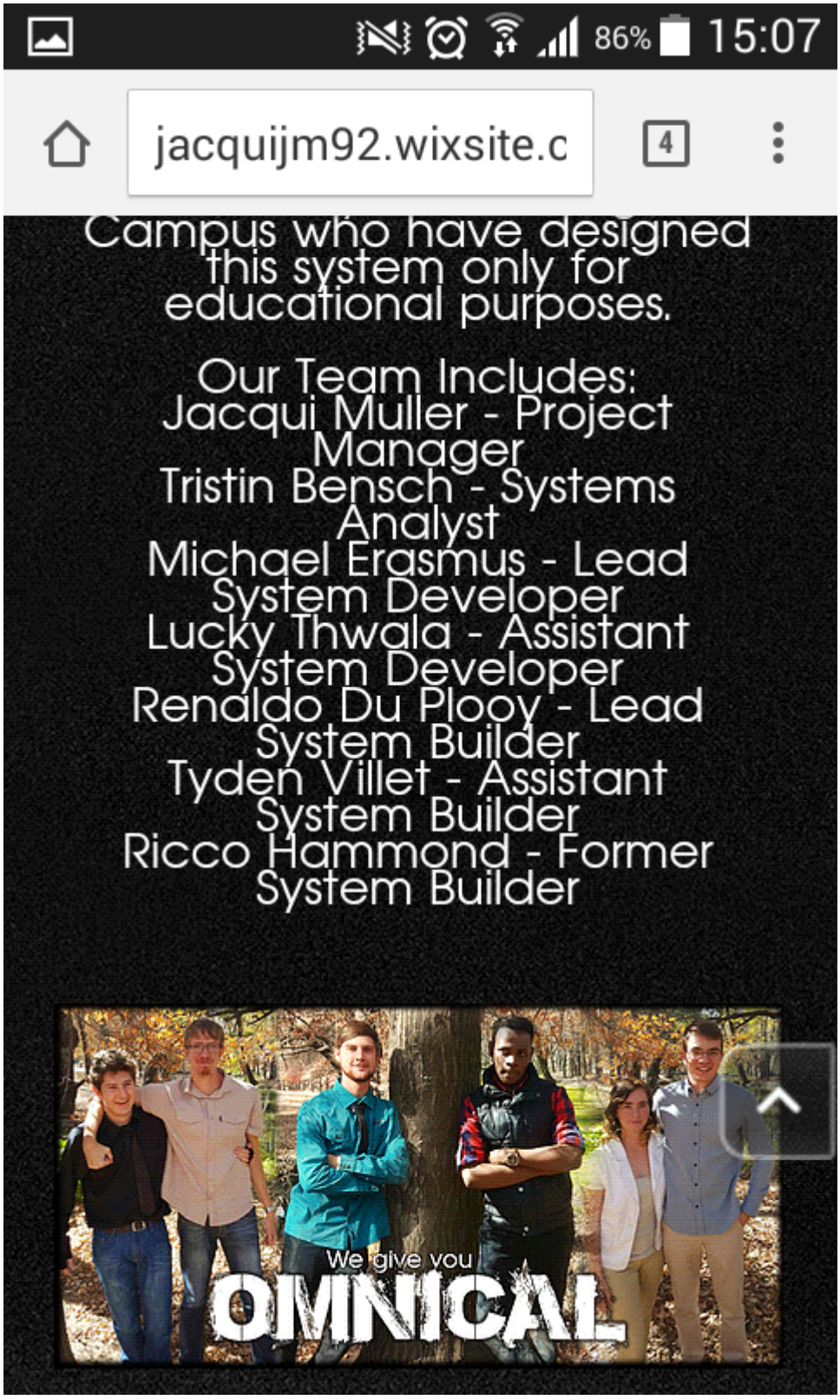
By: J Muller

By: J Muller

About Us

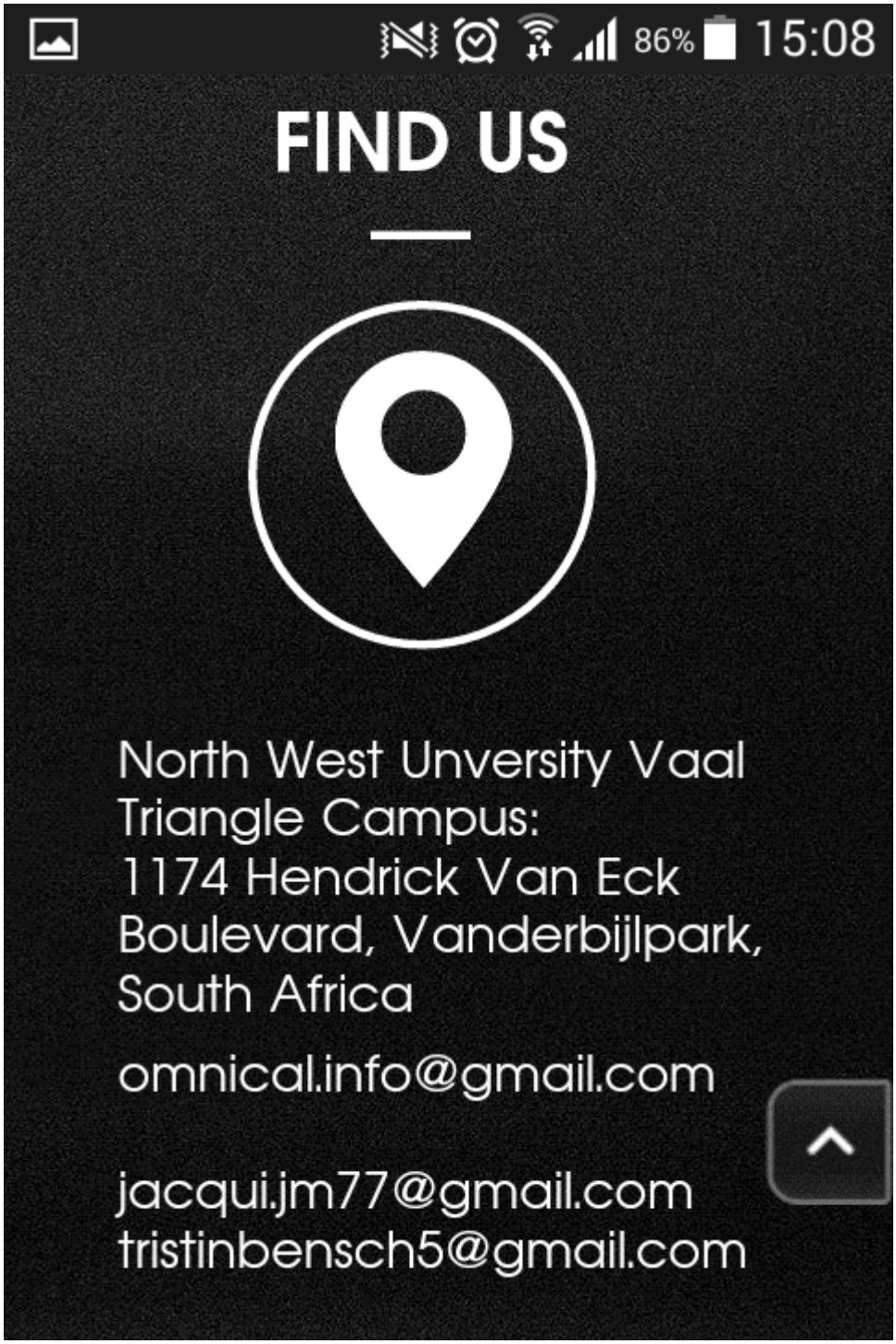


By: J Muller


86% 15:07
jacquijm92.wixsite.c
4
Campus who have designed this system only for educational purposes.
Our Team Includes:
Jacqui Muller - Project Manager
Tristin Bensch - Systems Analyst
Michael Erasmus - Lead System Developer
Lucky Thwala - Assistant System Developer
Renaldo Du Plooy - Lead System Builder
Tyden Villet - Assistant System Builder
Ricco Hammond - Former System Builder
We give you
OMNICAL

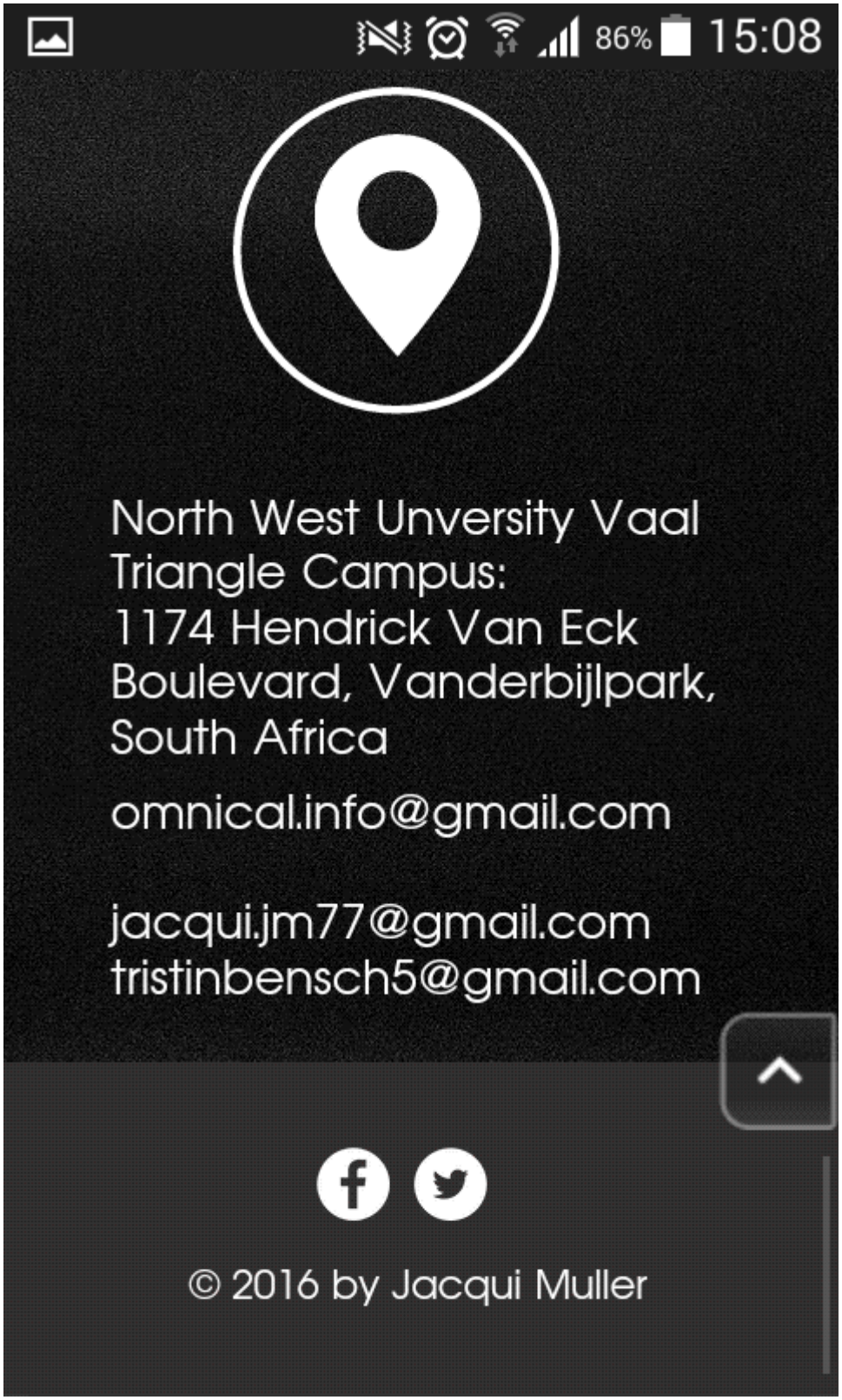
By: J Muller

Contact Us



By: J Muller

Social Media Links



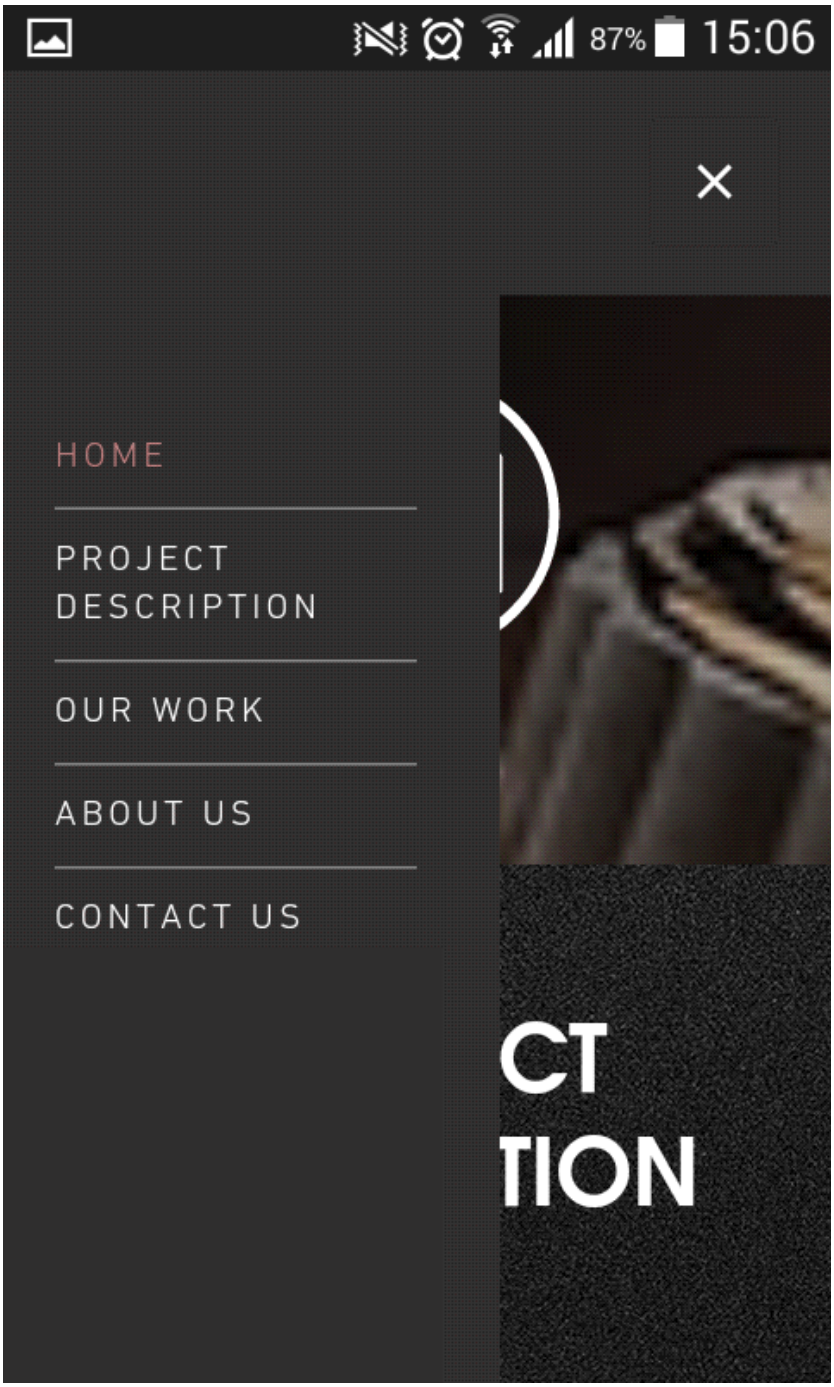
By: J Muller

Revised Desktop Online Input/ Output Screens

Desktop View

By: J Muller

Mobile View



By: J Muller

OBJECT ORIENTATED DESIGN

Refining Use Cases

Transforming the "Analysis" use Case to "Design" use cases

Refined Use Cases:

Send information to the database

User-Case Name:	Send information to the database.	User-Case Type Business requirements : Yes
User-Case ID	SI01	
Priority	High	
Source:	Requirement	
Primary Business Actor:	Anyone in need of a personal organizer	
Other Participating actors:	- Students - Working class - Sportsmen	
Other interested stakeholders:	-Designer : View Day Planner -Programmer: Create the event	
Description	This use-case describes the event of the designer and the programmer making use of this function in order to send information to the database for storage.	
Precondition	The information sent by the user must be sent through an account that has been created or is being created by the user	
Trigger	This Use case is only triggered when the administrator/User wants to save or change any information with the systems database	
Typical Course of event	Actor Action	System Response
	Step 1: The administrator/User login into the system **Step 3**: The user chooses one of the buttons by using the mouse or navigation arrows on the keyboard and **Step 2** the user press enter or right click on the	**Step 2:** The system will display the home screen page which will apply the user with choices of according to what the User desire to alter According to the option which the administrator/User has chosen

	mouse to choose the preferred option. **Step 5**: The user will click to the relevant fields they wish to change and enter or update their User demographic information or contact information if the user is done then he/she will enter the save button **Step 7:** The user will have to verify the information he entered if its correct and the system will verify any format error through validators and the user will click the (save) button if the information is correct. **Step 9:** A timetable will have to be automatically generated if the student is a BSc IT student after the database has been manipulated by the user	as the options will be indicated by buttons on the screen. **Step 4:** Once the user has chosen the preferred choices to alter, especially the update profile button to update or when the user has to register to the system. The user will be asked to enter the variables or to the fields he/she wants to update. **Step 6:** For every field that the user has to update the system will either accept or give the user a validation error if ever the format which the user has entered is in correct form or not. If the validation messages are still visible the user will not be able to continue updating the information **Step 8:** The information will be updated and the VT1 and CE1 use cases will be triggered as the events will have to be created by the system for the students. **Step 10:** The time-table will be available for the user to view
Alternate Course:	**Alt – Step 1:** If the user attempts to login the system without registering in the system, messages will be shown to the user that the information entered is incorrect since the system will not be recognizing the user. Any incorrect data entered by the user will not be accepted by the system and the user will be denied access to the system. **Alt – Step 2:** When the user has been denied the access to use the services offered the by OMNICAL, he or she must click on the (Register) button on the login screen so that the user can enter their user-information and be added to the database. **Alt-Step 4:** If the user attempts to enter invalid information the system will respond with validation text that will ask the user to enter the information in the correct format **Alt – step 9:** It is highly possible that the student has been registered to an unintended course or the student modules do not correspond with the enrolled curriculum of the student.	

	The system will review the details according to what information the system has received from the user and the time-table viewed will be according to the details given. If any information that has problem with the modules or curriculum, the student will have to consult the administration office and enquire about their details, especially IT students since the time-table will be automatically generated **Alt Step 10:** The OMNICAL team or system has added the MEET OUR TEAM form on the system which have the information which the user can send any queries or errors opposed by the system to the user. The queries will be directed to the database administrator of OMNICAL if any of the issues are related to the systems database
Conclusion	The conclusion of the use case happens when the user information/changes has been successfully updated into the database
Post condition	The user schedule has to be updated and the time-table of the user has to be generated.
Business rules	➢ User has to be a registered student with the entity ➢ User has to be registered on the OMNICAL system
Implementation constraints and specifications:	➢ The availability of the Use case to the student/User must be 24/7 ➢ The estimated time for the use case to be executed will be 8000 times at the beginning of every term and 500 times every month
Assumptions:	➢ The time-table can be seen by friends only by the allowance of the user ➢ The time-table might change according to the lecturer of admin ➢ Member can delete account ➢ If student is not an IT student the information entered manually will influence the correctness of the system
Open Issues	None

Receive information from the database

User-Case Name:	Receive information from the database.	User-Case Type Business requirements : Yes
User-Case ID	LO1	
Priority	High	
Source:	Requirement	
Primary Business Actor:	People interested in planning their daily lives.	
Other Participating actors:	-Students -Working class -Sportsmen	
Other interested stakeholders:	-Designer -Programmer: Create the event	
Description	This use-case describes the stakeholders' use of this function in order to receive information from the database to build or edit the current users table.	
Precondition	The information received from the database by the user must be sent through an account that has been created or is being created by the user	
Trigger	This Use case is only triggered when the administrator/User wants to view or make use of any information with the systems database	
Typical Course of event	Actor Action	System Response
	Step 1 – The administrator/User login into the system	**Step 2:** The system will receive information from the database to verify the information and then display the home screen page which will apply the user with choices of according to what the User desire to alter
	Step 3: The user chooses one of the buttons by using the mouse or navigation arrows on the keyboard and **Step 2** the user press enter or right click on the mouse to choose the preferred option.	According to the option which the administrator/User has chosen as the options will be indicated by buttons on the screen. **Step 4:** Any necessary info is loaded. **Step 6:** For every field that the user wants to view, the system will

	Step 5: The relevant information based on the users selection is loaded from the database	either accept changes or decline them
Alternate Course:	**Alt – Step 1:** If the user attempts to login the system without registering in the system, messages will be shown to the user that the information entered is incorrect since the system will not be recognizing the user. Any incorrect data entered by the user will not be accepted by the system and the user will be denied access to the system. **Alt – Step 2:** When the user has been denied the access to use the services offered the by OMNICAL, he or she must click on the (Register) button on the login screen so that the user can enter their user-information and be added to the database. **Alt-Step 4:** If the user attempts to receive invalid information the system will respond with an error that will ask the user to request the information in the correct format	
Conclusion	The conclusion of the use case happens when the user information/changes has requested form information to be uploaded from the database	
Post condition	The user schedule has to be uploaded and the time-table of the user has to be generated.	
Business rules	➢ User has to be a registered student with the entity ➢ User has to be registered on the OMNICAL system	
Implementation constraints and specifications:	➢ The availability of the Use case to the student/User must be 24/7 ➢ The estimated time for the use case to be executed will be 8000 times at the beginning of every term and 500 times every month	
Assumptions:	➢ The user is logged into the system.	
Open Issues	None	

Create Event

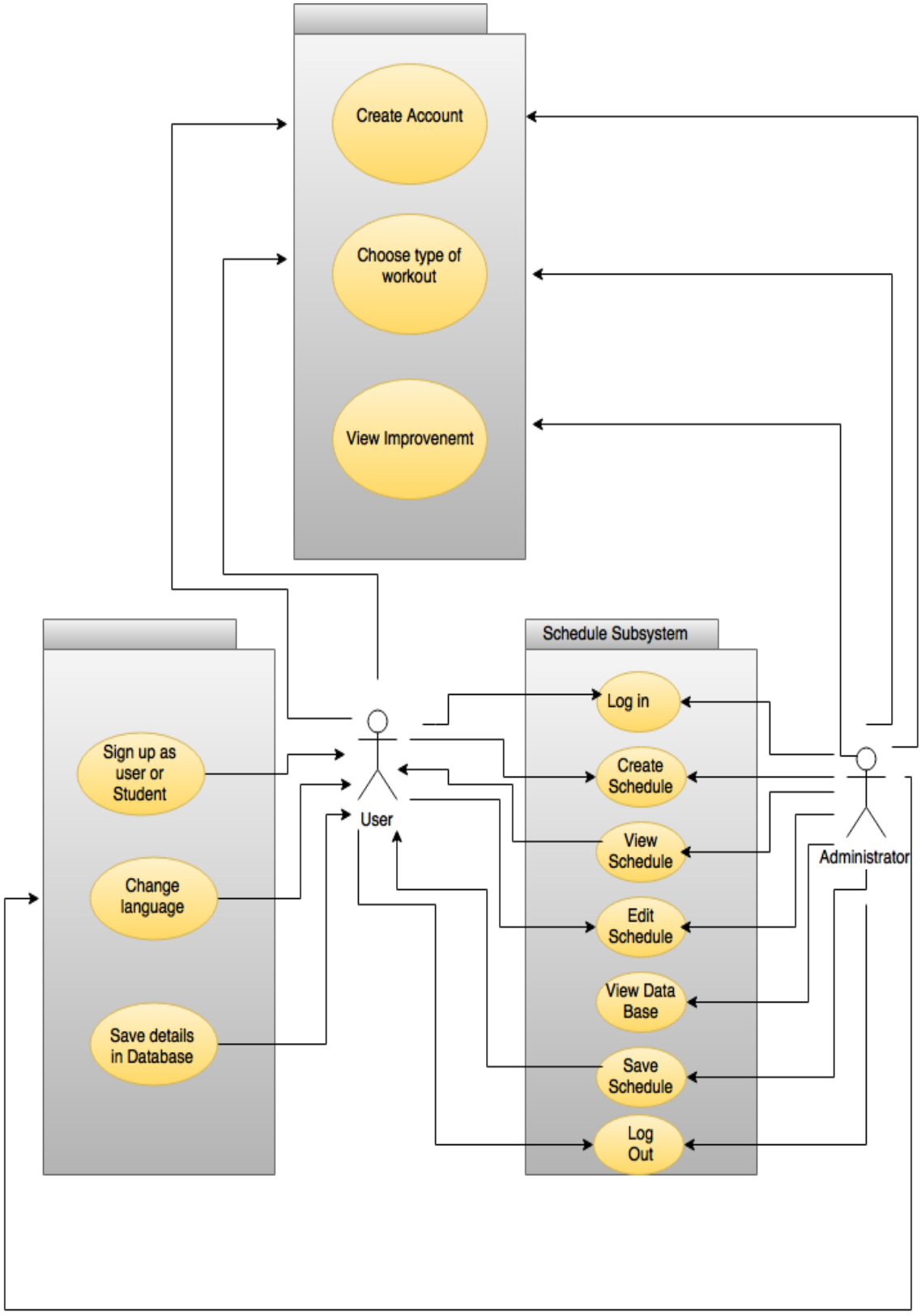
Use-Case Name:	Create Event	
Use-Case ID:	CE1	
Priority:	High	
Soure:	Create Event Requirement	
Primary Business Actor:	Registered User	
Description:	This use-case deals with the creation/scheduling of an event in OmniCal to be updated to a user's planned timetables and events.	
Precondition:	The user must be registered and logged in.	
Trigger:	This use-case is initiated when the user views a day in the planner and selects the option to create an event.	
Typical Course of Events:	**Actor Action** **Step 1:** The registered and logged-in user selects a day to view. **Step 3:** The user types out the information to be stored in the schedule and saves/creates it.	**System Response** **Step 2:** The selected day schedule is displayed. **Step 4:** The information is saved to the database (see "Send Infor to Database" use-case).
Alternate Courses:	n/a	
Conclusion:	An event has been stored in the specific day's schedule for the user.	
Post-condition:	The event must be sent to the database.	
Business Rules:	n/a	
Implementation Constraints and Specifications:	GUI to be provided for the user and a database for storing all necessary information.	
Assumptions:	The event will be stored appropriately in the database (separate use-case).	
Open Issues:	1. An event must be stored in a manner that includes a time of day. 2. It is undecided yet whether or not more than one event can be created for the same day and time.	

View Timetable

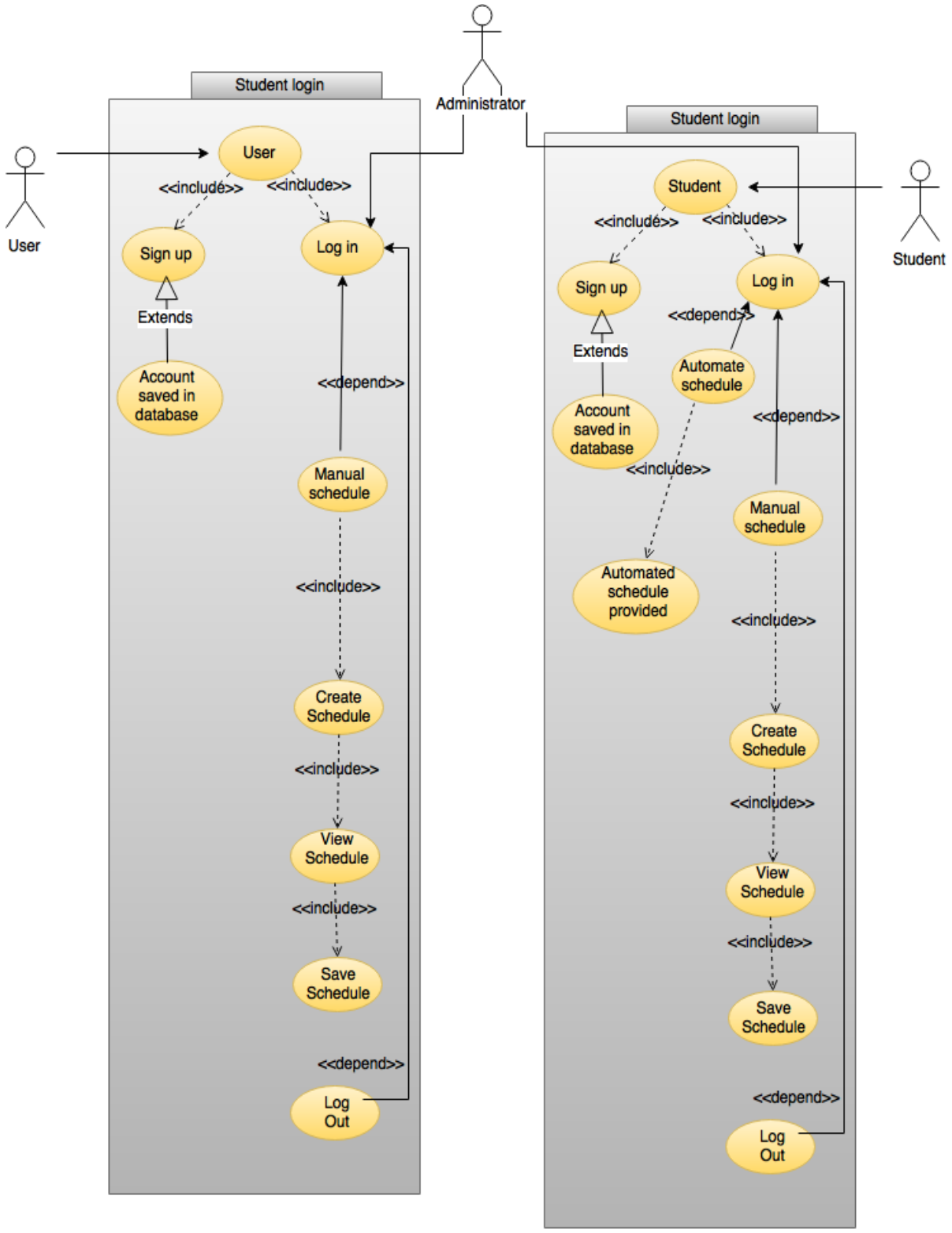
Use-Case Name:	View Timetable	
Use-Case ID:	VT1	
Source:	Database	
Primary Business Actor	Registered User	
Description:	The use-case send a specific request of a registered client to the database of OmniCal that then will send the clients current days activities for display.	
Precondition:	Only one week will be displayed in the view timetable option.	
Trigger:	The use-case will initiate when the log in use-case is complete and the view timetable option is selected.	
Typical Courses of Events:	**Actor Action** **Step 1:** The logged in user selects view day option.	**System Response** **Step 2:** Database loads the logged in users details sends the info to be displayed by the GUI.
Alternate Courses:	n/a	
Conclusion:	The view timetable is viewed or edited by the user.	
Post condition:	The View timetable use-case receives data from the database and the user.	
Business rules:	n/a	
Implementations Constraints and Specifications:	GUI to display the users schedule and database to store the users data.	
Assumptions:	The users schedule will be customised by the view timetable use-case and stored in the database in the appropriate users data.	
Open issues:	If the view timetable only displays one day at a time the users is going to take a long time to edit the schedule to there needs.	

By: J Muller & LT Thwala

Use Case Diagram



Use Case Diagram (Cont.)



Modeling class interactions, behaviors, and states that support the use-case scenario

Step 1:

Interface classes	Controller classes	Entity classes
Log in	UpdateInfo	View calendar
Home	UpdateAdminInfo	Announcements
EditModule	UpdateProfile	View Statistics
Register		View Timetable
View data		Language
Timetable		
License Agreement		
Preferences		

Step 2:

Upon further Inspection of the refined use-case's it can be seen that there are no discrepancies for attributes between the new and old use-case's.

Step 3:

Task 1 – Partial summary of OmniCal use-case behaviors

Behaviors	Automated/manual	Class type
Receive timetable	Automated	Entity
Check if empty	Automated	Entity
Display Timetable	Automated	Entity
Send user preference	Manual	Entity
Send user info	Manual	Controller
Send user timetable	Manual	Interface
Send user language	Manual	Entity
Retrieve user preferences	Automated	Interface
Retrieve user info	Automated	Controller
Retrieve user timetable	Automated	Interface
Retrieve user language	Automated	Entity
Retrieve password	Automated	Controller
Retrieve username	Automated	Controller

By: TB Bensch

Step 3:

Class behaviors and responsibilities

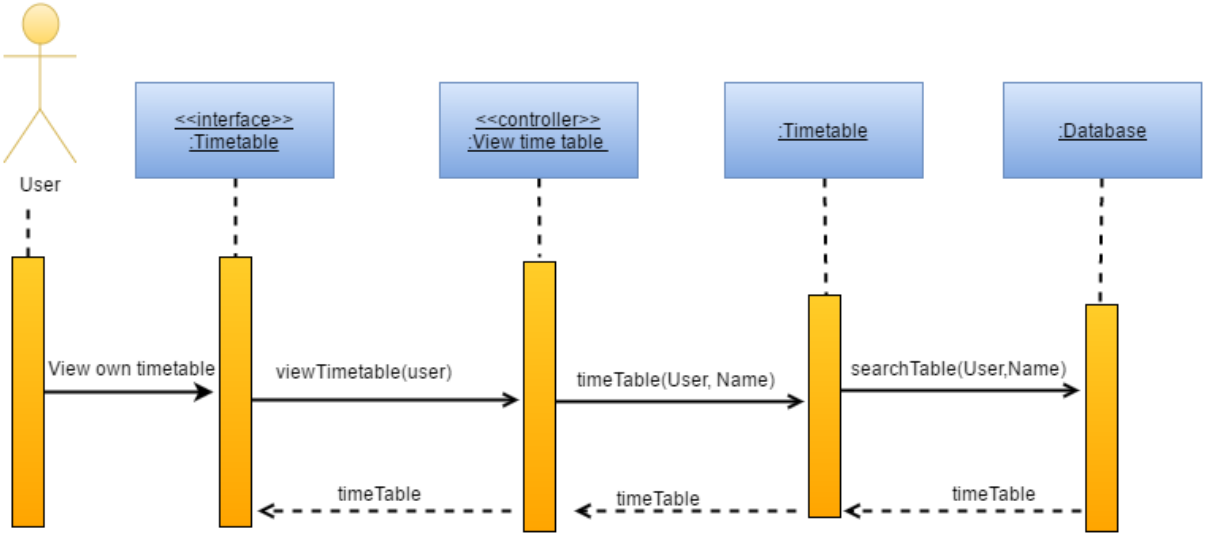
CRC Cards

Object Name: View Timetable	
Sub Object:	
Super Object: View	
Behaviors and Responsibilities	**Collaborators**
Receive user timetable	Edit Timetable
Check for emty timetable	
Display user timetable	

Object Name: Send User info	
Sub Object:	
Super Object: update	
Behaviors and Responsibilities	**Collaborators**
Send user preference	Edit Timetable
Send user info	user preference
Send user timetable	
Send user language	

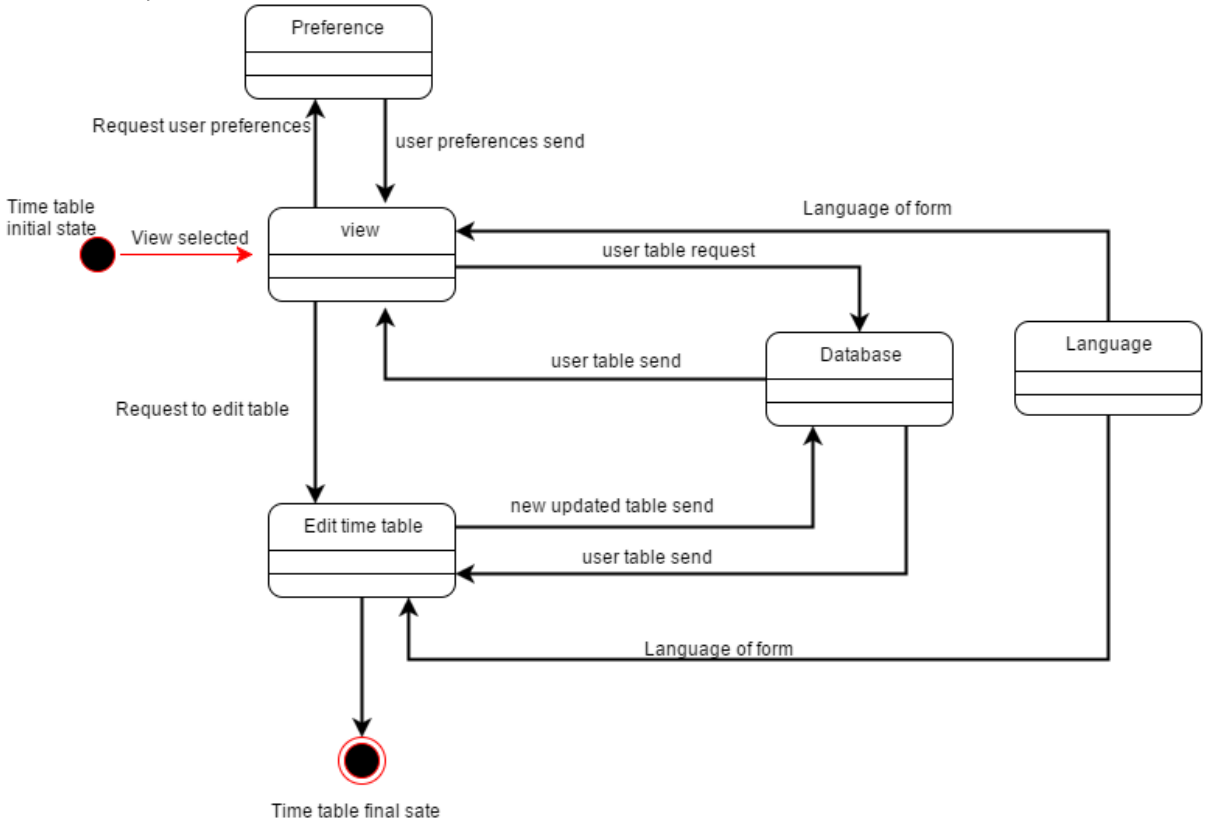
Object Name: Receive user info	
Sub Object:	
Super Object: retrieve	
Behaviors and Responsibilities	**Collaborators**
Retrieve user preference	Log in
Retrieve user info	edit timetable
Retrieve user timetable	edit preference
Retrieve user language	
Retrieve password	
Retrieve username	

Sequence Diagram



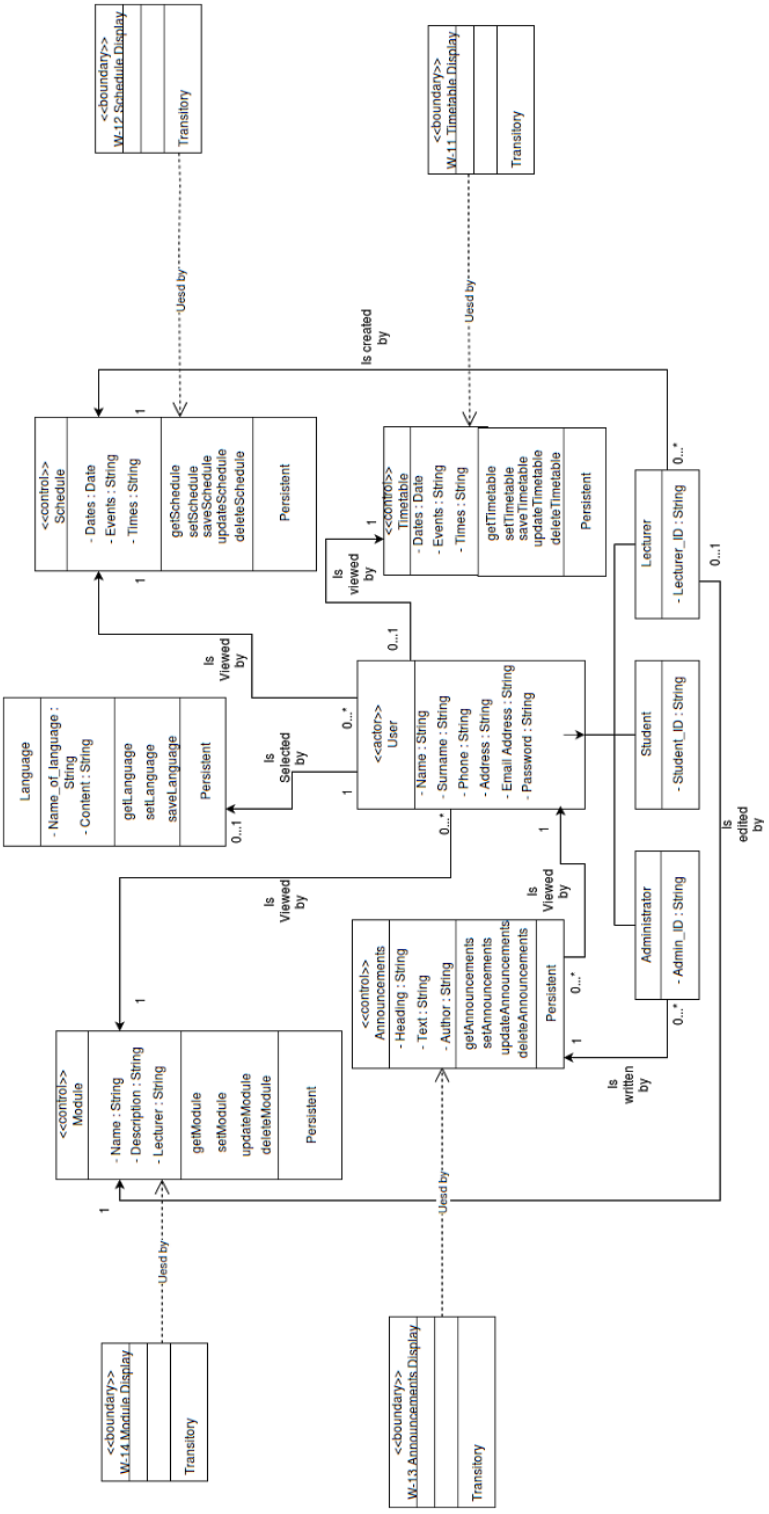
Step 4:

Model Object States



By: R Du Plooy

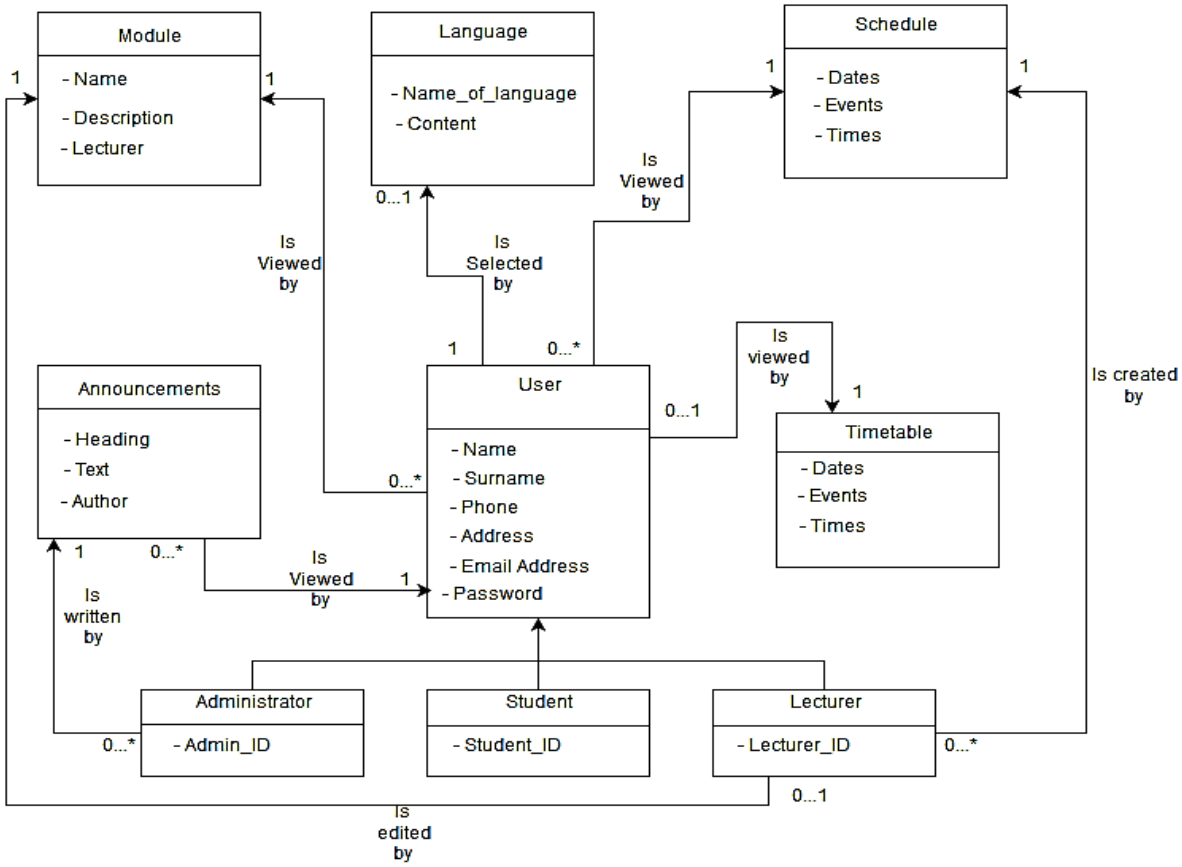
Updating the OM to reflect the implementation environment



By: MC Erasmus & TR Villet

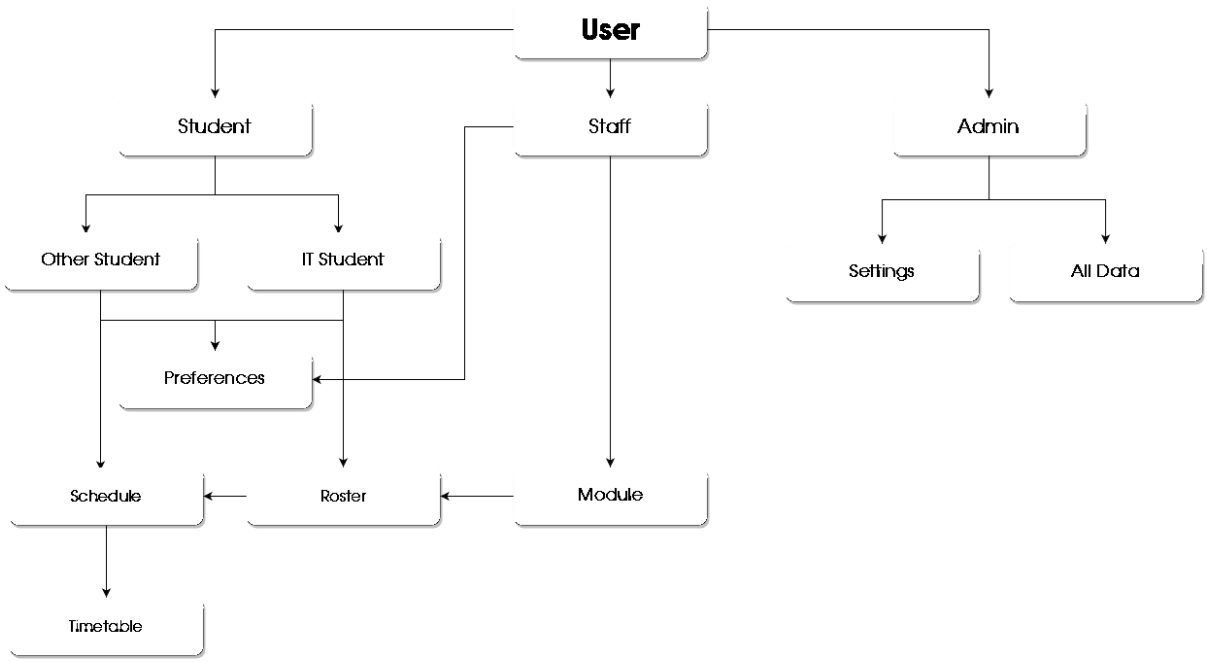
SECOND SEMESTER FINAL DOCUMENTATION

Adapter Pattern



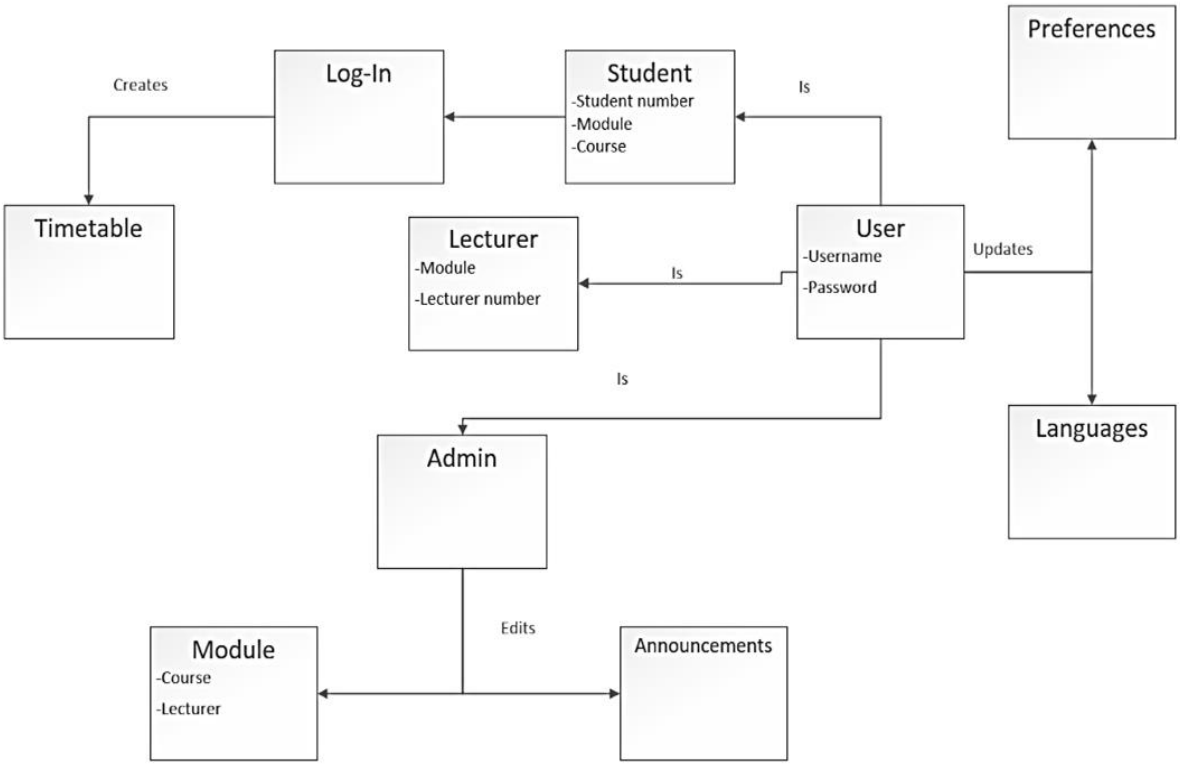
By; TR Villet

Organisational Pattern



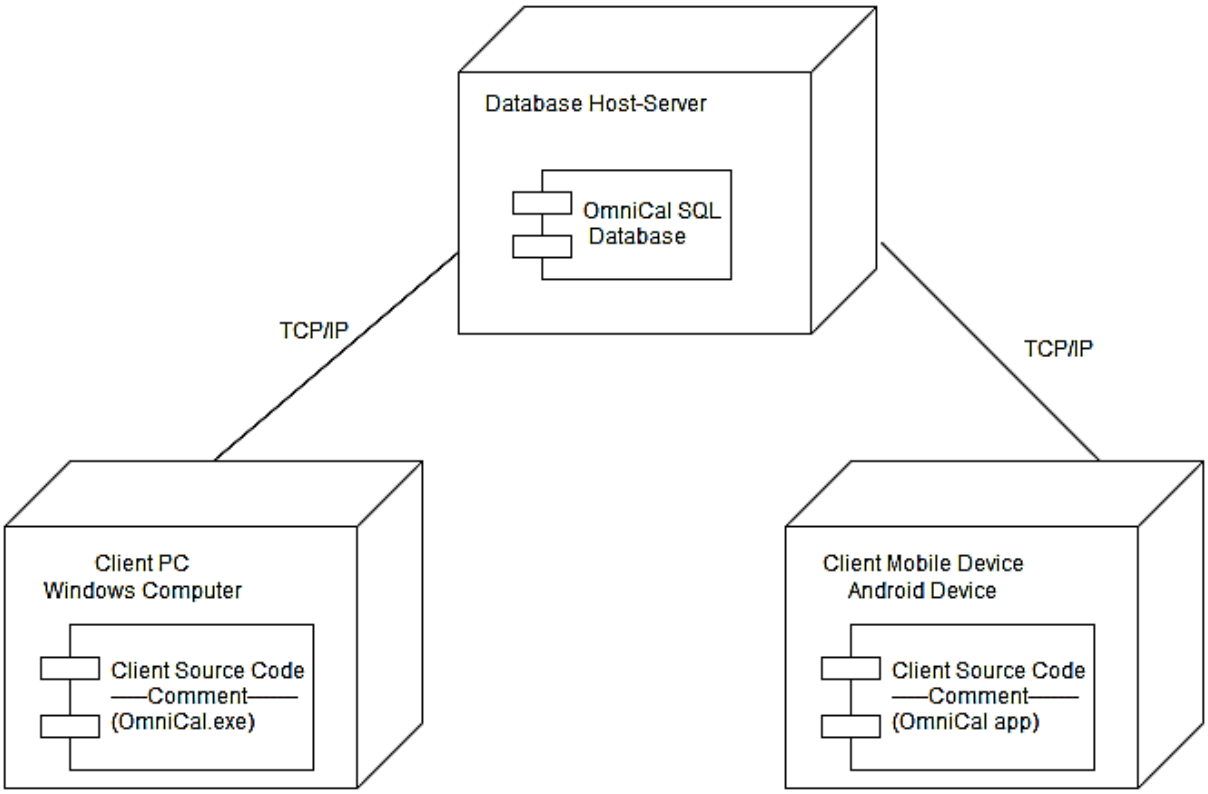
By: J Muller

Strategy Pattern



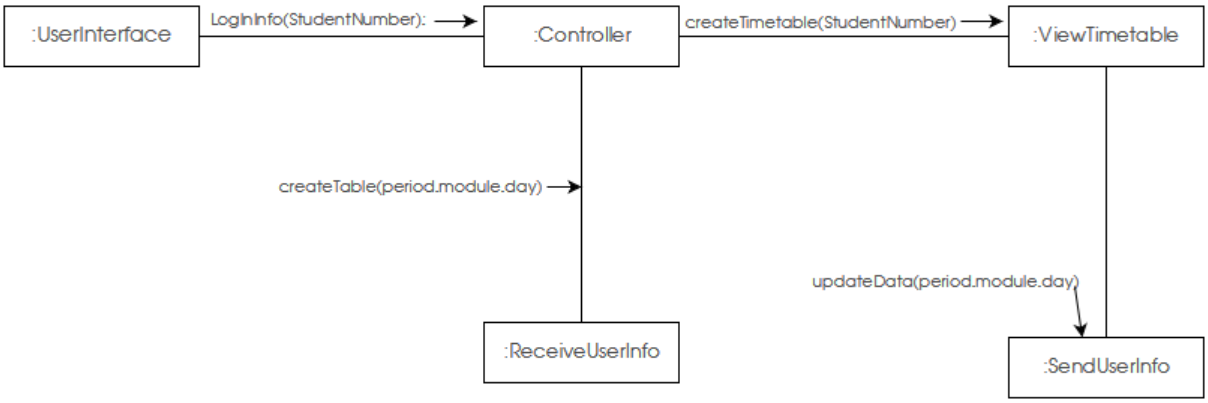
By: TB Bensch

Deployment diagram



By: MC Erasmus

Communication diagram



By: TB Bensch

SYSTEM IMPLEMENTATION AND CONSTRUCTION

Implementation Phase

1. CONDUCT SYSTEM TEST

Test:	Result
Network	☑
Databases	☑
In-house software	☑
Existing software	☑

Test data:

- Data stored in database in various tables:
  - admin
  - announcements
  - calendarevents
  - course
  - gui
  - module
  - period
  - preferences
  - requests
  - roster
  - schedule
  - staff
  - student

- Data stored in textfiles for various language options
  - omnical licence agreement Afrikaans
  - omnical licence agreement English
  - omnical licence agreement Zulu
  - frmAnnouncements English
  - frmAnnouncements Afrikaans
  - frmAnnouncements Zulu
  - frmEditModules English
  - frmEditModules Afrikaans
  - frmEditModules Zulu
  - frmHome Afrikaans
  - frmHome English
  - frmHome Zulu
  - frmLicenseAgreement English
  - frmLicenseAgreement Afrikaans
  - frmLicenseAgreement Zulu
  - frmLogin English
  - frmLogin Afrikaans
  - frmLogin Zulu
  - frmMeetTheTeam Afrikaans
  - frmMeetTheTeam English
  - frmMeetTheTeam Zulu
  - frmPreferences English
  - frmPreferences Afrikaans

    - frmPreferences Zulu
    - frmRegisteration Afrikaans
    - frmRegisteration English
    - frmRegisteration Zulu
    - frmSettings Afrikaans
    - frmSettings English
    - frmSettings Zulu
    - frmStatistics Afrikaans
    - frmStatistics English
    - frmStatistics Zulu
    - frmTimetable Afrikaans
    - frmTimetable English
    - frmTimetable Zulu
    - frmUpdateAdmin Afrikaans
    - frmUpdateAdmin English
    - frmUpdateAdmin Zulu
    - frmUpdateProfile Afrikaans
    - frmUpdateProfile English
    - frmUpdateProfile Zulu
    - frmViewCalendar Afrikaans
    - frmViewCalendar English
    - frmViewCalendar Zulu
    - frmViewTimetable Afrikaans
    - frmViewTimetable English
    - frmViewTimetable Zulu

- Problems and issues
    - n/a
- Problems revealed during testing
    - n/a
- Verification of system operation
    - The parts of the system that operate with test data operate correctly

By: J Muller

2. PREPARE CONVERSION PLAN

Databases to be installed:

- OmniCal database
    - admin
    - announcements
    - calendarevents
    - course
    - gui
    - module
    - period
    - preferences
    - requests
    - roster
    - schedule
    - staff
    - student

End-user Training:

- End-user training will be simple and easy to accomplish, through the use of integrated help pages and an intuitive query page, allowing users to ask questions or report problems to a helpful OmniCal employee.

Conversion Strategy:

- The OmniCal system will most likely be converted using the Staged conversion strategy, being released in versions, where each new version will be converted to in parallel with each previous version slowly being replaced. This allows us, as the OmniCal team, time to discover all problems and issues with any new releases before they become major problems.

By: TR Villet

3. INSTALL DATABASES

OmniCal Database installed	☑

By: J Muller

4. TRAIN USERS (Help File – User Manual)

Please see user manual after documentation.

By: TB Bensch

5. CONVERT TO NEW SYSTEM

This link shows how the system has been converted from the pre-prototype to everything we have now
https://www.youtube.com/watch?v=4W69fecnRIM

By: J Muller

CONSTRUCTION PHASE

1. BUILD AND TEST NETWORKS

Any good web design placements include an assessing strategy, and the termination of the assessing plan. To test the web design, you have to design the examination web of nature, and afterwards craft the test network. An assessing nature can encompass one lab or several labs. You can craft assorted labs to examine specific constituents inside the web design.

There are a number of hardware components and mechanisms that demand to be encompassed inside the testing network. A number of normal hardware components and mechanisms that are to be included inside the examination nature for OmniCal are:

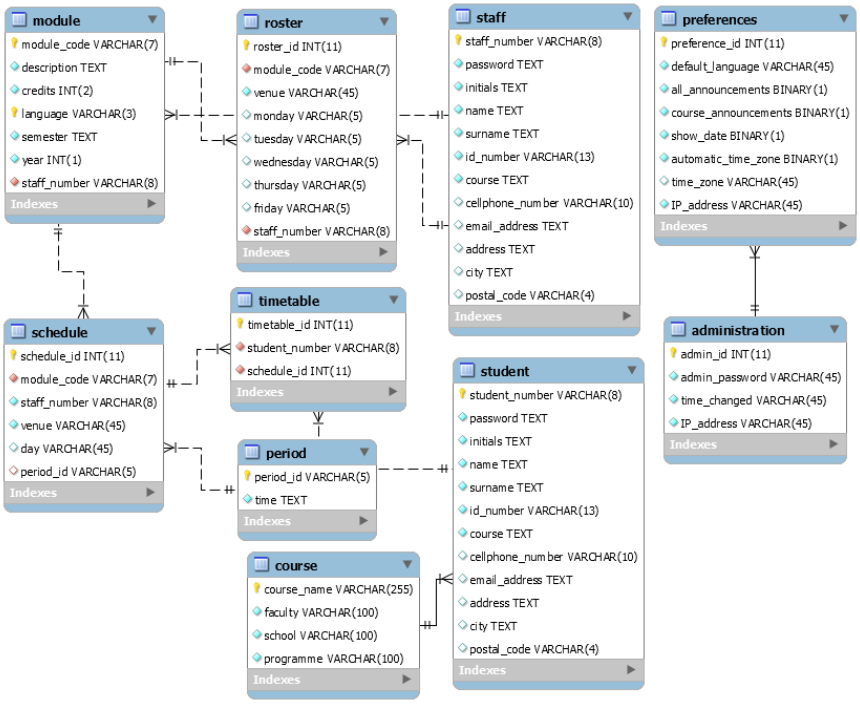
- Network Adapters
- USB Adapters for User input
- Mouse Devices
- Printers
- Monitor
- Touchpad

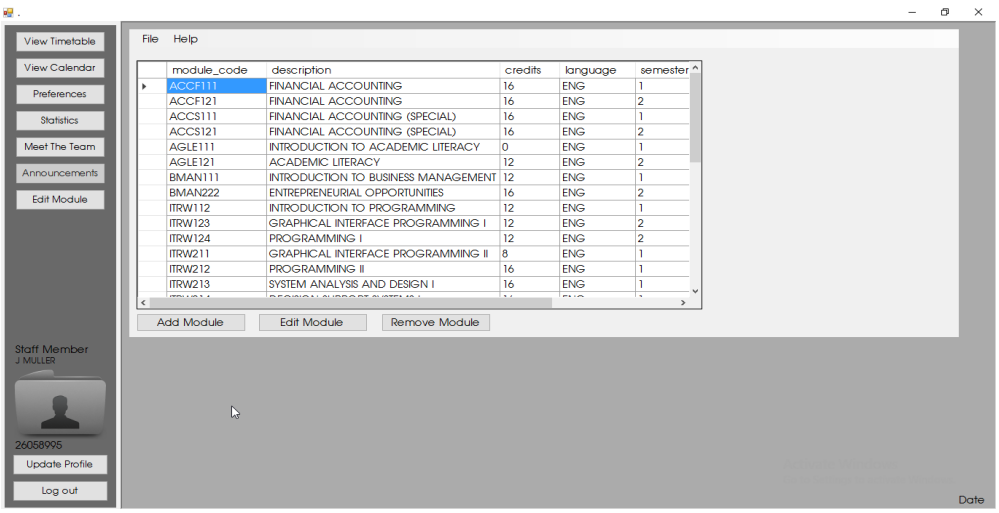
Networking Services:

- Windows Server 2003 networks is an option depending on who will be using or giving the OmniCal service to the Student
- And MySQL server

By: LT Thwala

2. BUILD AND TEST DATABASES





These links show that the data does come from the database and respective tables

https://www.youtube.com/watch?v=TUOSBU1gHqU
https://www.youtube.com/watch?v=l9HKVqxC6gY

By: J Muller

3. INSTALL AND TEST NEW SOFTWARE
   - New Software

Software	Installed	Tested	Result
MySQL	☑	☑	☑
PDF Package	☑	☑	☑
Printer Package	☑	☑	☑

By: J Muller

4. WRITE AND TEST NEW PROGRAMS
   https://www.youtube.com/watch?v=4W69fecnRIM
   Use this link to reference our program. Also, make a note that this phase is still incomplete

By: J Muller, TB Bensch, R Du Plooy, LT Thwala, MC Erasmus

SYSTEM SUPPORT & OPERATIONS

Systems Support and Operations Checklist

TASKS		**METHOD USED**
System maintenance	✓	-
Validate problems	✓	Code
Benchmark Program	✓	Build the program
Study and Debug the program	✓	Step through
Test the Program	✓	Build and run program
System Recovery	✓	**Reload from Google Drive as well as through the admin option on the system**
Technical support	✓	-
Routinely observations	✓	Through admin on the system
User satisfaction surveys	✓	The survey is found on the website (it is updated regularly) and users are made aware about the survey through social media
Training	✓	In form of the help file and user manual
Log enhancements ideas	✓	This is done through email and Wix notifications
System enhancement	✓	-
Analyse Enhancements request	✓	Through email notifications as well as direct Wix notifications
Quick Fix methods	✓	Administration will demonstrate quick fixes directly on the system
Recover existing physical system	✓	This can be done through the Google Drive
System Obsolescence	-	n/a

By: R Du Plooy & J Muller