OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:	Tuesday:	Wednesday:	Thursday:
1-2: ITRW212 12 G01	1-2: ACCS111-1 12 G01	1-2: ITRW214 3-103	1-2: ACCS111-1 12 G05
5-6: ITRW211 3 103	5-6: ITRW212 3 103	3-4: ITRW214 3-103	3-4: ITRW211 9B G02
9-10: WVNS211 25 G08	7-8: ITRW212 3 103	5-6: ITRW213 3 103	5-6: ITRW212 3 103
11-12: ITRW213 25 G01		7-8: ITRW213 3-103	7-8: ITRW212 3 103

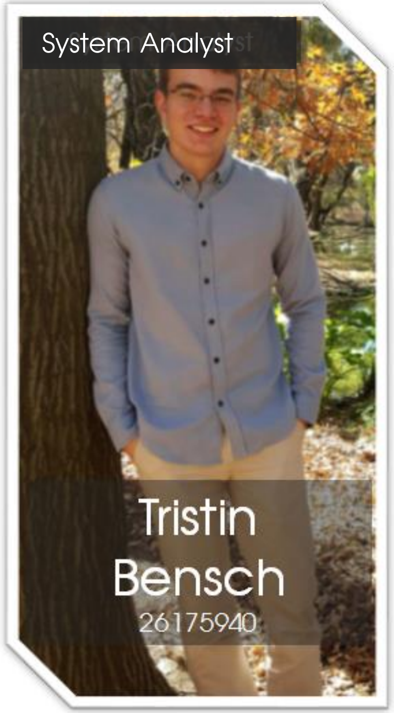
MEET THE TEAM

OMNICAL GALLERY

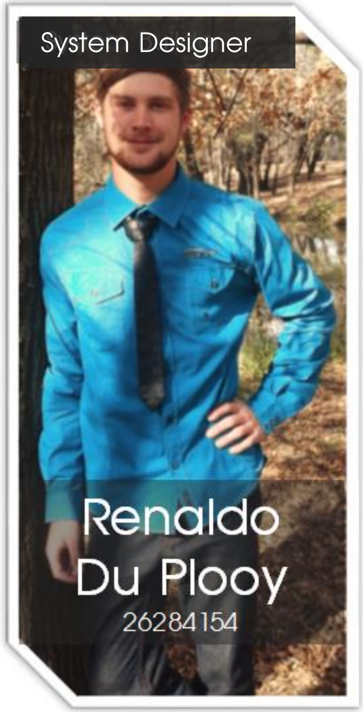
OmniCal Stakeholders:

Project Manager

Jacqueline Muller 26058995


System Analyst

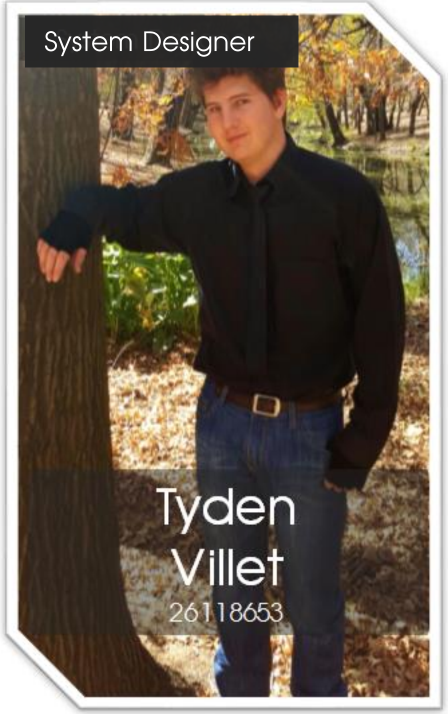
Tristin Bensch 26175940


System Designer

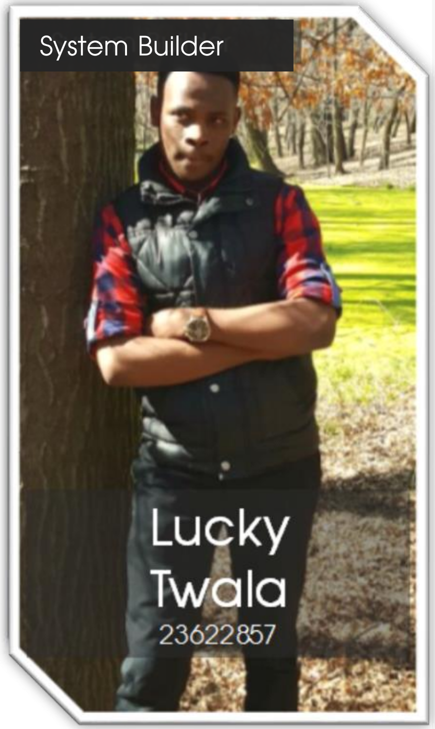
Renaldo Du Plooy 26284154

System Builder

Michael Erasmus 26438585

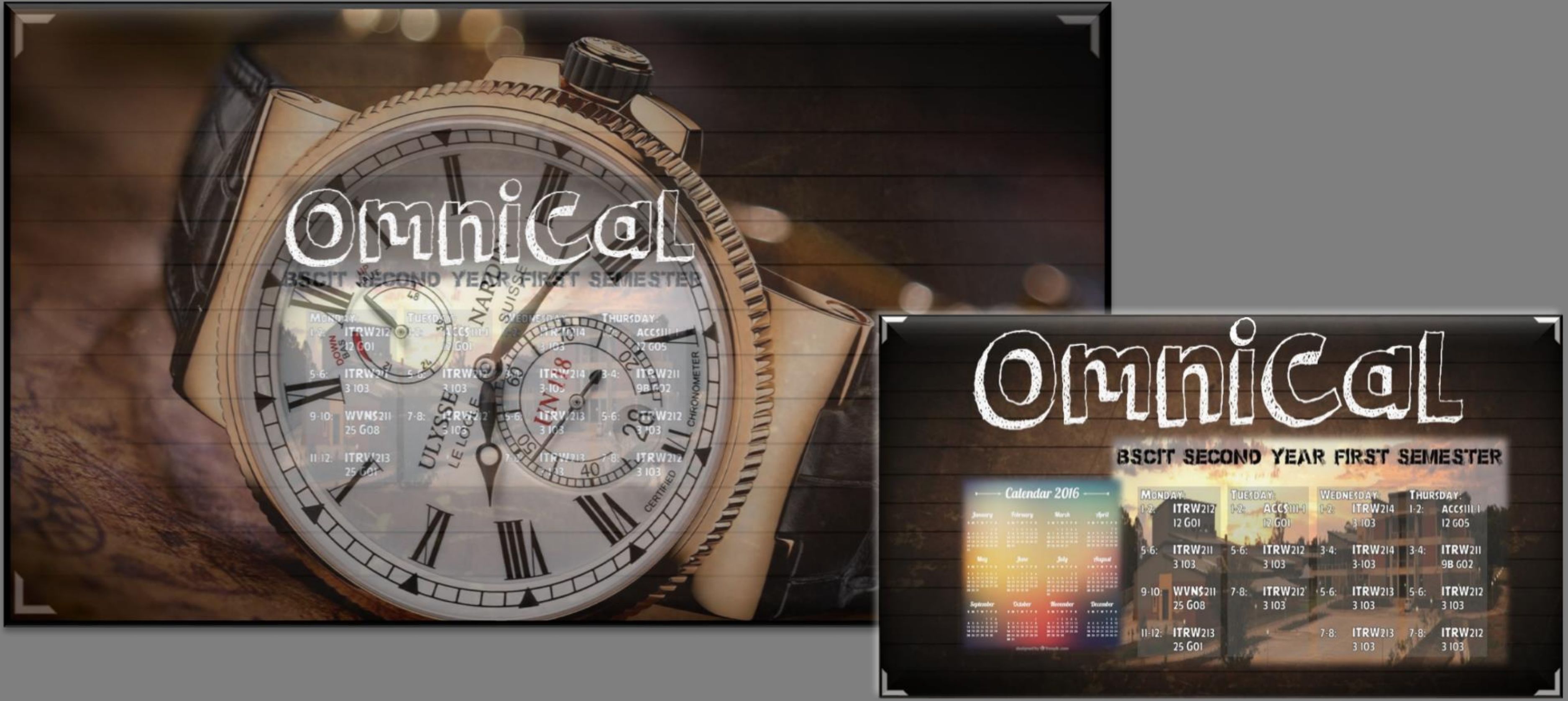

System Designer

Tyden Villet 26118653


System Builder

Lucky Twala 23622857

PROJECT DESCRIPTION



OmniCaL

BSCIT SECOND YEAR FIRST SEMESTER

Monday:	Tuesday:	Wednesday:	Thursday:
1-2: ITRW212 12 G01	1-2: ACCS111-1 12 G01	1-2: ITRW214 3 103	1-2: ACCS111-1 12 G05
5-6: ITRW211 3 103	5-6: ITRW212 3 103	3-4: ITRW214 3-103	3-4: ITRW211 9B G02
9-10: WVNS211 25 G08	7-8: ITRW212 3 103	5-6: ITRW213 3 103	5-6: ITRW212 3 103
11-12: ITRW213 25 G01		7-8: ITRW213 3 103	7-8: ITRW212 3 103

PROJECT DESCRIPTION

OmniCal is a powerful day-planning, academic scheduling system, designed with students and lecturers in mind.

OmniCal Allows the user to organize and put their time to better use, through arranging events and even organizing their timetables, by providing them with this easy-to-use and easy-to-understand system. 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)

OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2.	ITRW212 12 G01	1-2:	ACCS111-1 12 G01	1-2:	ITRW214 3-103	1-2:	ACCS111-1 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3-103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3-103	7-8:	ITRW212 3 103

REFINING USE CASE

OmniCaL

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2:	ITRW212 12 G01	1-2:	ACCS111 17 G01	1-2:	ITRW214 3 103	1-2:	ACCS111 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3 103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3 103	7-8:	ITRW212 3 103

USE CASE LIST

The list below indicates all of the use cases that OmniCal makes use of:

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
Log in	Data is now accessible and editable
Log out	Data is no longer accessible and editable
Sign up	Information is created in order for log in to occur
Create event	Data is created to be sent to the database.
View timetable	Data is accessed and can be viewed
If admin	All data is accessible and editable
If student	Data for only that student can be accessed and edited
If worker	Data for only that worker can be accessed and edited
Help file	No data used
If English	Data is translated to English
If Afrikaans	Data is translated to Afrikaans
If Zulu	Data is translated to Zulu
If Sesotho	Data is translated to Sesotho
Close program	All data is released

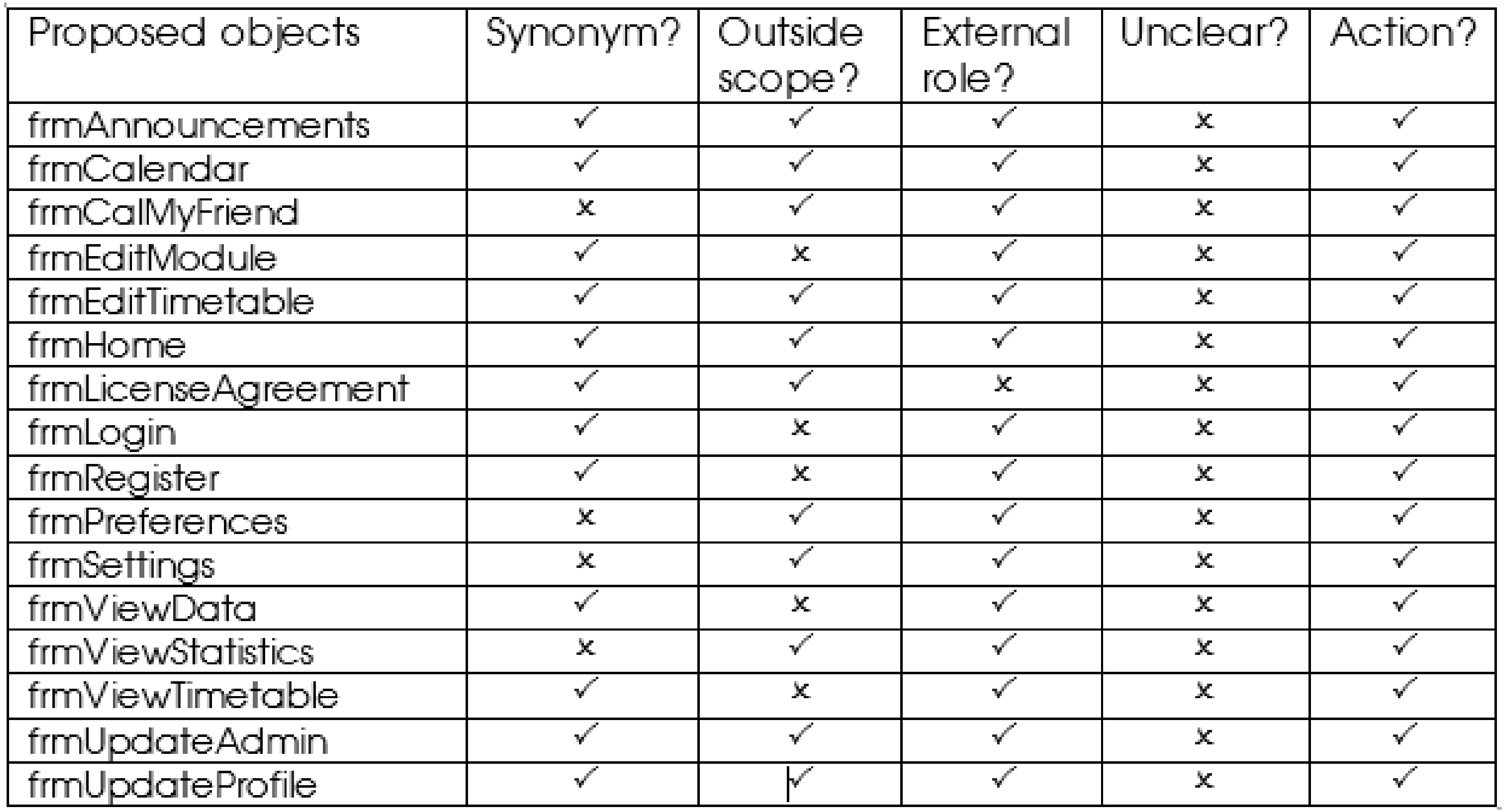
REFINED USE CASE LIST

The list below indicates all of the refined use cases that OmniCal makes use of:

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

SELECTION OF PROPOSED OBJECTS

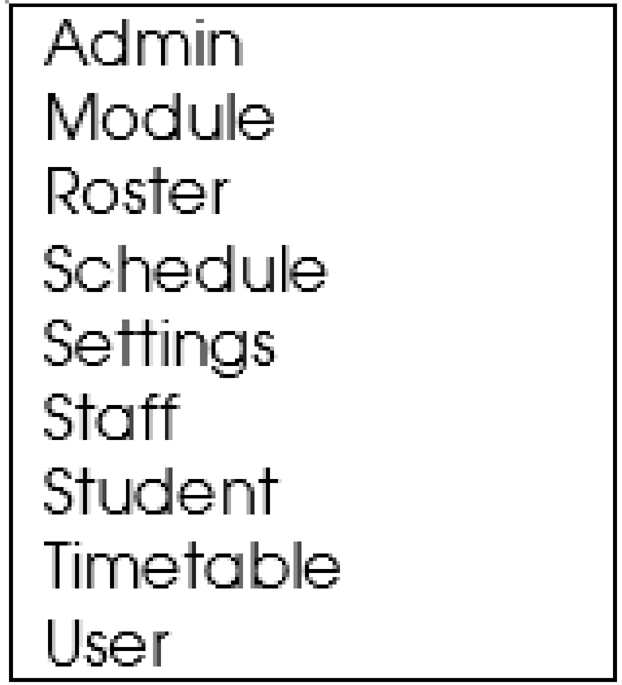
The table below shows the evaluation of proposed objects



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

PROPOSED OBJECTS

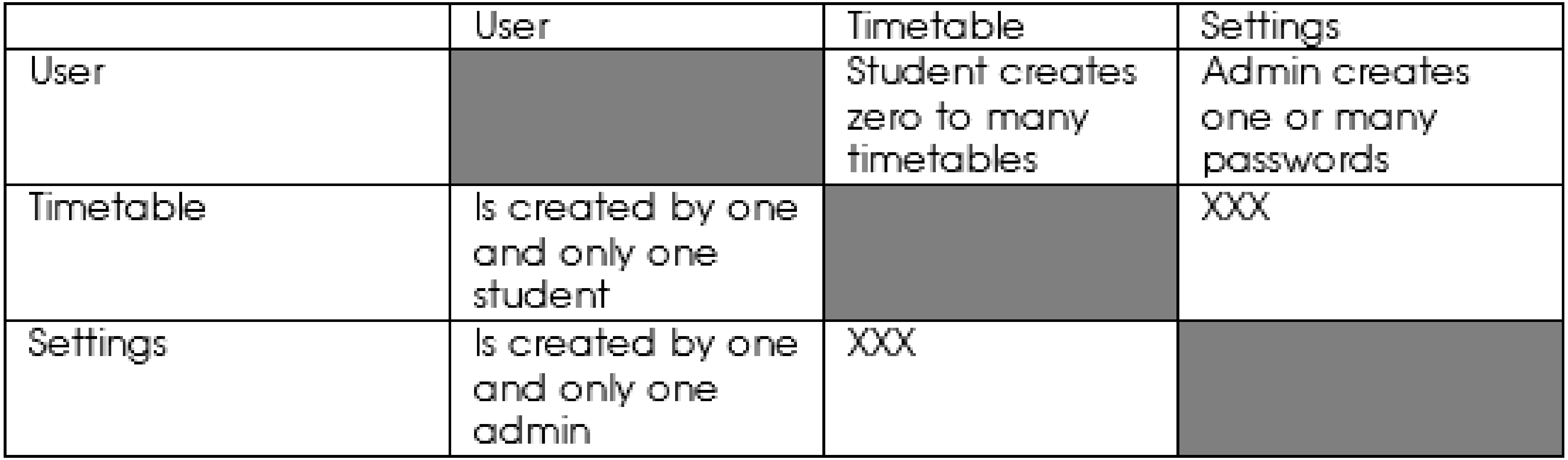
The table below shows the proposed objects



Admin
Module
Roster
Schedule
Settings
Staff
Student
Timetable
User

ASSOCIATION MATRIX

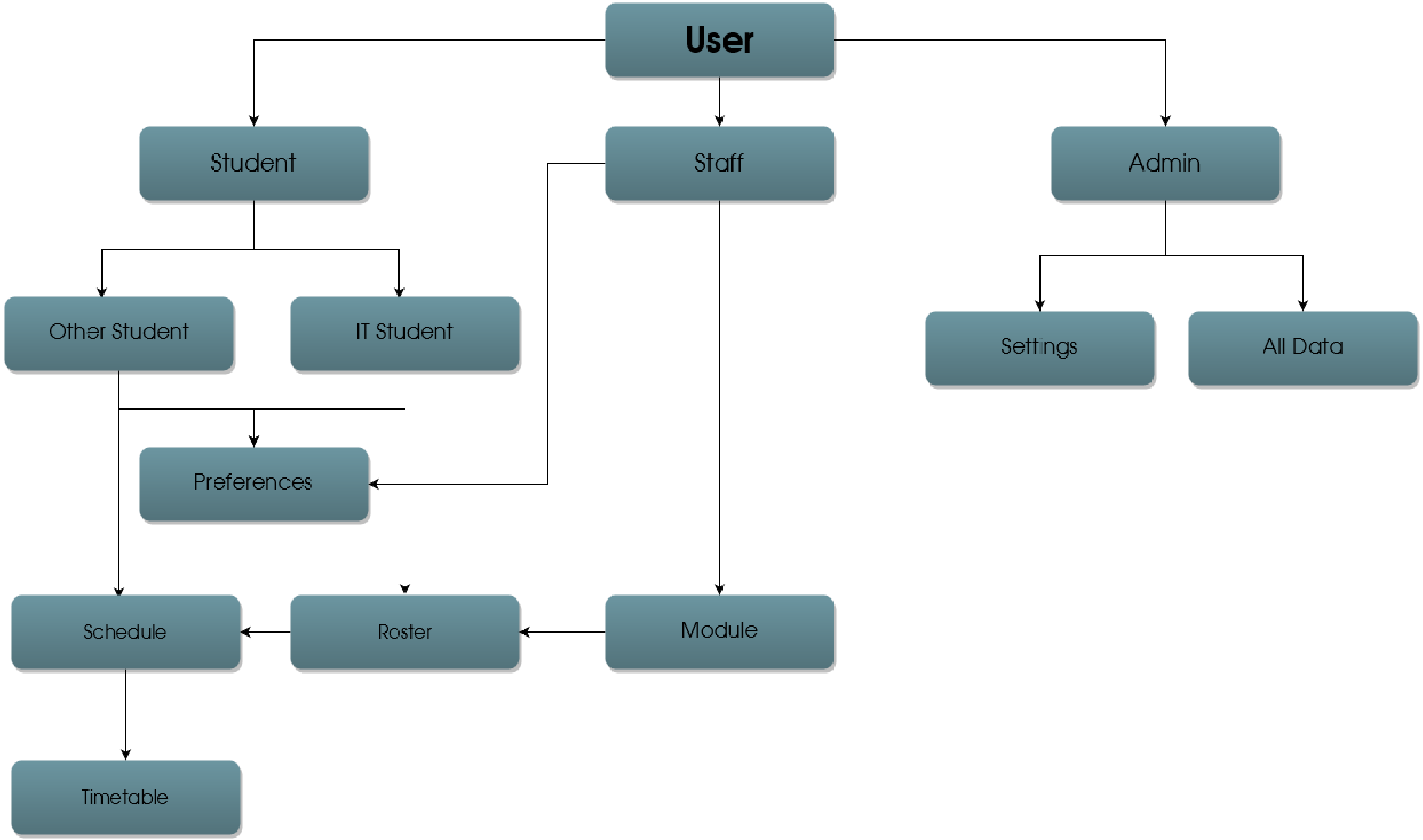
The table below shows the multiplicity and associations between objects



	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	

GENERALISATION - SPECIALISATION

The diagram below shows the generalisation and specialisation relationships between our objects



AGGREGATION RELATIONSHIPS

The list below depicts only aggregate relationships between the objects

- Timetable contains Schedule and is therefore*, an aggregate relationship.*
- The relationship between Preferences and IT Students/ Other Students / staff is an *aggregate relationship.*
- The relationship between Roster and Schedule is *aggregate.*
- Likewise, the relationship between Staff and Module is *aggregate.*
- The relationship between Module and Roster is an *aggregate relationship.*

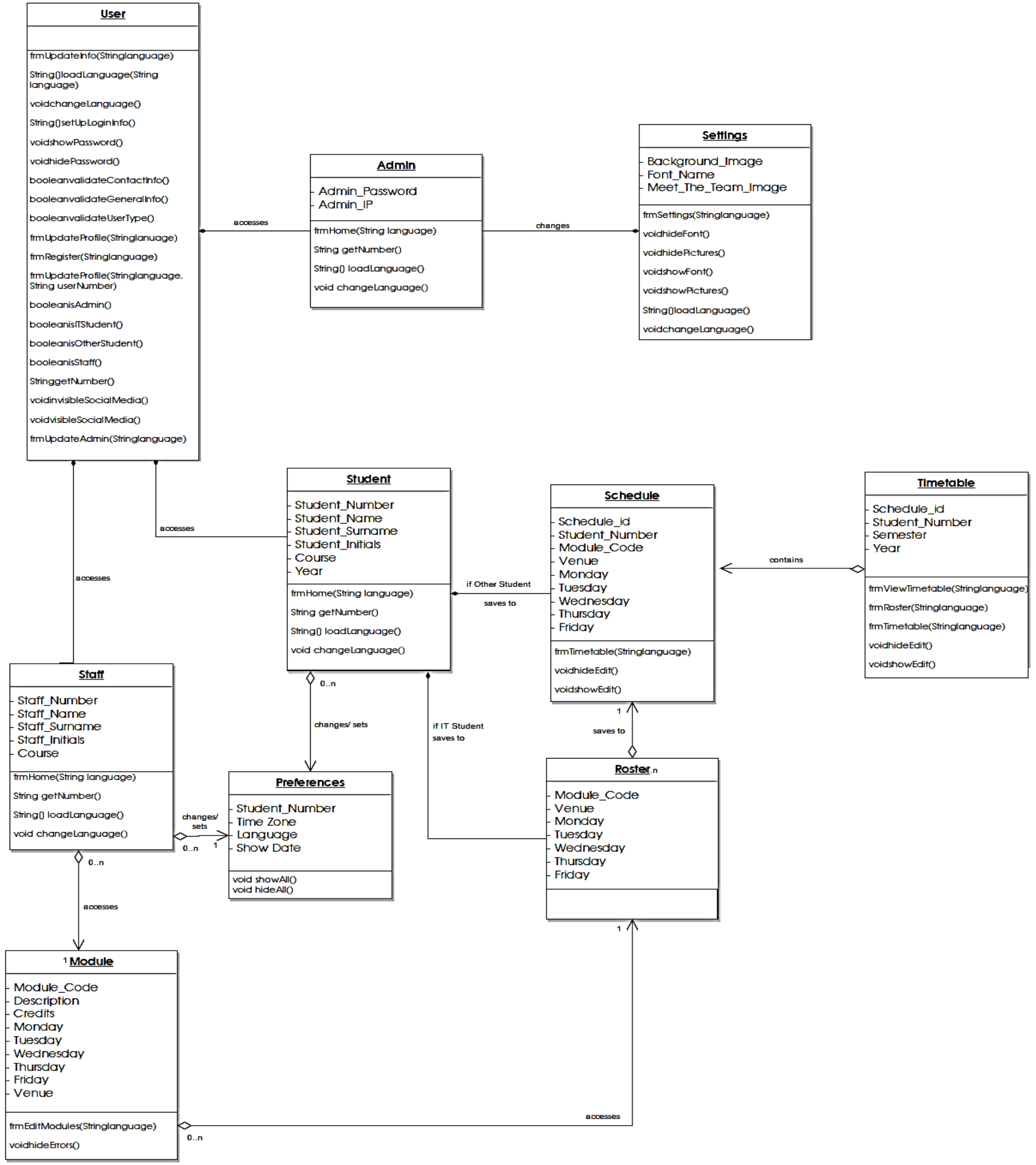
COMPOSITION RELATIONSHIPS

The list below depicts only composition relationships between the objects

- User contains Student, Staff and Admin. These subclasses each contain their own objects. This represents a *composition relationship*
- Other student and student also represent a *composition relationship*
- The relationship between Other Student and Schedule is a *composition relationship.*
- Likewise, IT Student and Roster also represent a *composition relationship*
- The relationship between Schedule and Timetable are *composite.*
- The relationships between Admin and Settings/ All data are *composite.*

CLASS DIAGRAM

The diagram below shows the classes that we make use of as well as their interactions



User

frmUpdateInfo(Stringlanguage)
String()loadLanguage(String language)
voidchangeLanguage()
String()setUpLoginInfo()
voidshowPassword()
voidhidePassword()
booleanvalidateContactInfo()
booleanvalidateGeneralInfo()
booleanvalidateUserType()
frmUpdateProfile(Stringlanuage)
frmRegister(Stringlanguage)
frmUpdateProfile(Stringlanguage. String userNumber)
booleanisAdmin()
booleanisITStudent()
booleanisOtherStudent()
booleanisStaff()
StringgetNumber()
voidinvisibleSocialMedia()
voidvisibleSocialMedia()
frmUpdateAdmin(Stringlanguage)

Admin

- Admin_Password
- Admin_IP

frmHome(String language)
String getNumber()
String() loadLanguage()
void changeLanguage()

Settings

- Background_Image
- Font_Name
- Meet_The_Team_Image

frmSettings(Stringlanguage)
voidhideFont()
voidhidePictures()
voidshowFont()
voidshowPictures()
String()loadLanguage()
voidchangeLanguage()

Student

- Student_Number
- Student_Name
- Student_Surname
- Student_Initials
- Course
- Year

frmHome(String language)
String getNumber()
String() loadLanguage()
void changeLanguage()

Schedule

- Schedule_id
- Student_Number
- Module_Code
- Venue
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday

frmTimetable(Stringlanguage)
voidhideEdit()
voidshowEdit()

Timetable

- Schedule_id
- Student_Number
- Semester
- Year

frmViewTimetable(Stringlanguage)
frmRoster(Stringlanguage)
frmTimetable(Stringlanguage)
voidhideEdit()
voidshowEdit()

Staff

- Staff_Number
- Staff_Name
- Staff_Surname
- Staff_Initials
- Course

frmHome(String language)
String getNumber()
String() loadLanguage()
void changeLanguage()

Preferences

- Student_Number
- Time Zone
- Language
- Show Date

void showAll()
void hideAll()

Roster

- Module_Code
- Venue
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday

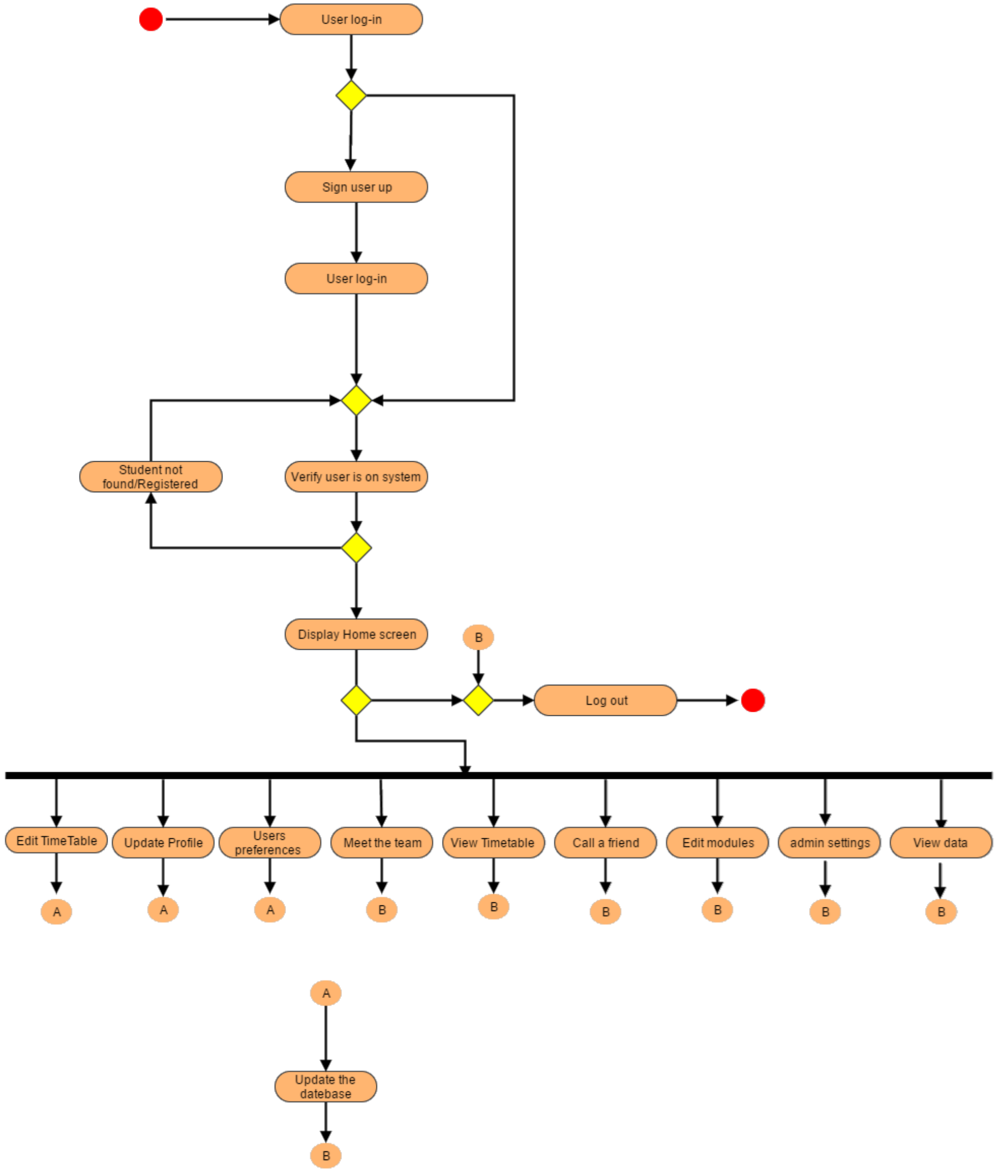
Module

- Module_Code
- Description
- Credits
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Venue

frmEditModules(Stringlanguage)
voidhideErrors()

ACTIVITY DIAGRAM

The diagram below shows the objects and activities that our system uses



ANALYSIS VS DESIGN

OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2:	ITRW212 12 G01	1-2:	ACCS111-1 12 G01	1-2:	ITRW214 3 103	1-2:	ACCS111-1 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3-103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3 103	7-8:	ITRW212 3 103

SYSTEM DESIGN VS SYSTEM ANALYSIS

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)

SYSTEM DESIGN VS SYSTEM ANALYSIS (Continued)

3. Requirement Analysis	3b When conducting the physical design and integration phase of system design, the following 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	c. Decision Analysis (for software and services)	Due to the improvement in problem analysis, requirements need 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)

SYSTEM DESIGN VS SYSTEM ANALYSIS (Concluded)

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 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	e. Design (and integration)	N/A

SYSTEM DESIGN

(PHYSICAL DESIGN)

SYSTEM DESIGN (PHYSICAL DESIGN)

- OmniCal uses a Model - Driven Approach which is comprised of:
  - **Modern structured design:** a system design technique that decomposes the system's processes into manageable components

  - **Information engineering:** Information Engineering models are pictures that illustrate and synchronize the system's data and processes.

  - **Prototyping:** a small-scale, incomplete, but working sample of a desired system

  - **Object-oriented:** techniques are used to refine the object requirements definitions identified earlier during analysis, and to define design specific objects

12 POINT PLAN TO 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:

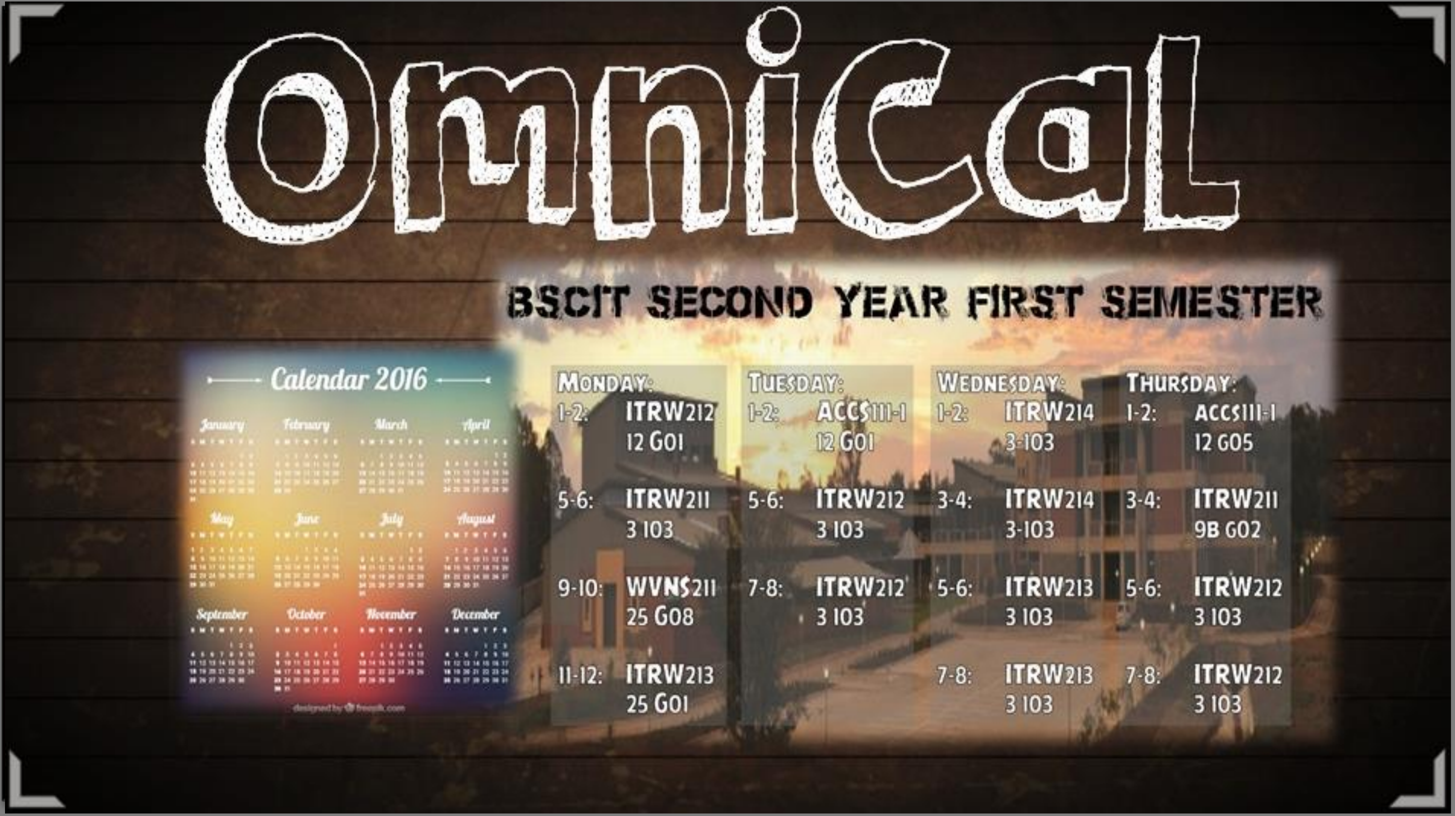
1. Add a registration process
2. Add a home screen
3. Add a “View Timetable Out Of Editor” option
4. Add a “View Statistics” option
5. Add a “User Preferences” option
6. Add an “Edit Modules” option for staff members and limit their access
7. Add a “Meet The Team” option
8. Add a “View in Calendar” option
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

OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2:	ITRW212 12 G01	1-2:	ACCS111-1 12 G01	1-2:	ITRW214 3-103	1-2:	ACCS111-1 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3-103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3-103	7-8:	ITRW212 3 103

APPLICATION ARCHITECTURE



OmniCal

Calendar 2016

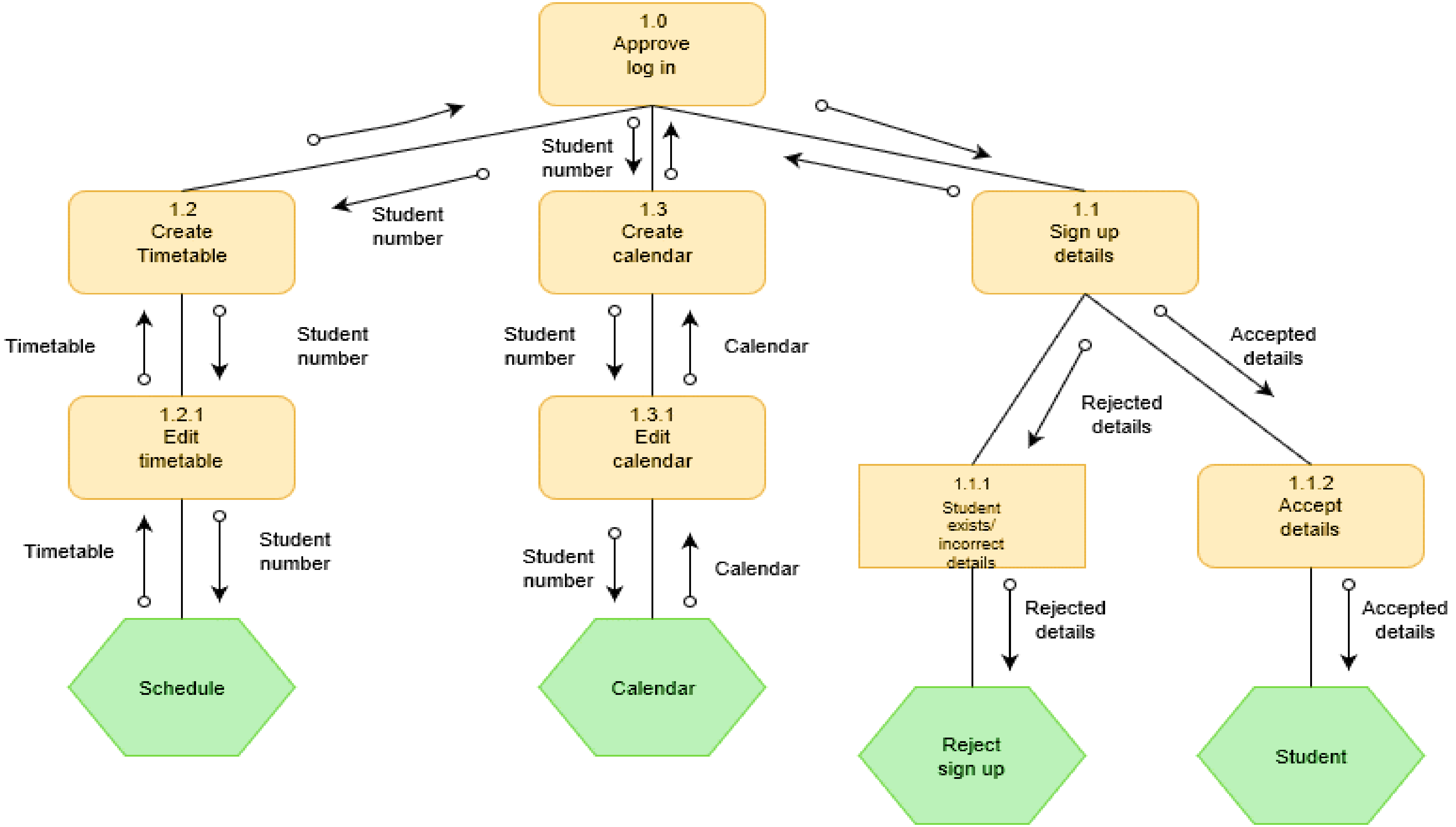
January February March April May June July August September October November December

BSCIT SECOND YEAR FIRST SEMESTER

Monday:	Tuesday:	Wednesday:	Thursday:
1-2: ITRW212 12 G01	1-2: ACC$111-1 12 G01	1-2: ITRW214 3-103	1-2: ACC$111-1 12 G05
5-6: ITRW211 3 103	5-6: ITRW212 3 103	3-4: ITRW214 3-103	3-4: ITRW211 9B G02
9-10: WVN$211 25 G08	7-8: ITRW212 3 103	5-6: ITRW213 3 103	5-6: ITRW212 3 103
11-12: ITRW213 25 G01		7-8: ITRW213 3 103	7-8: ITRW212 3 103

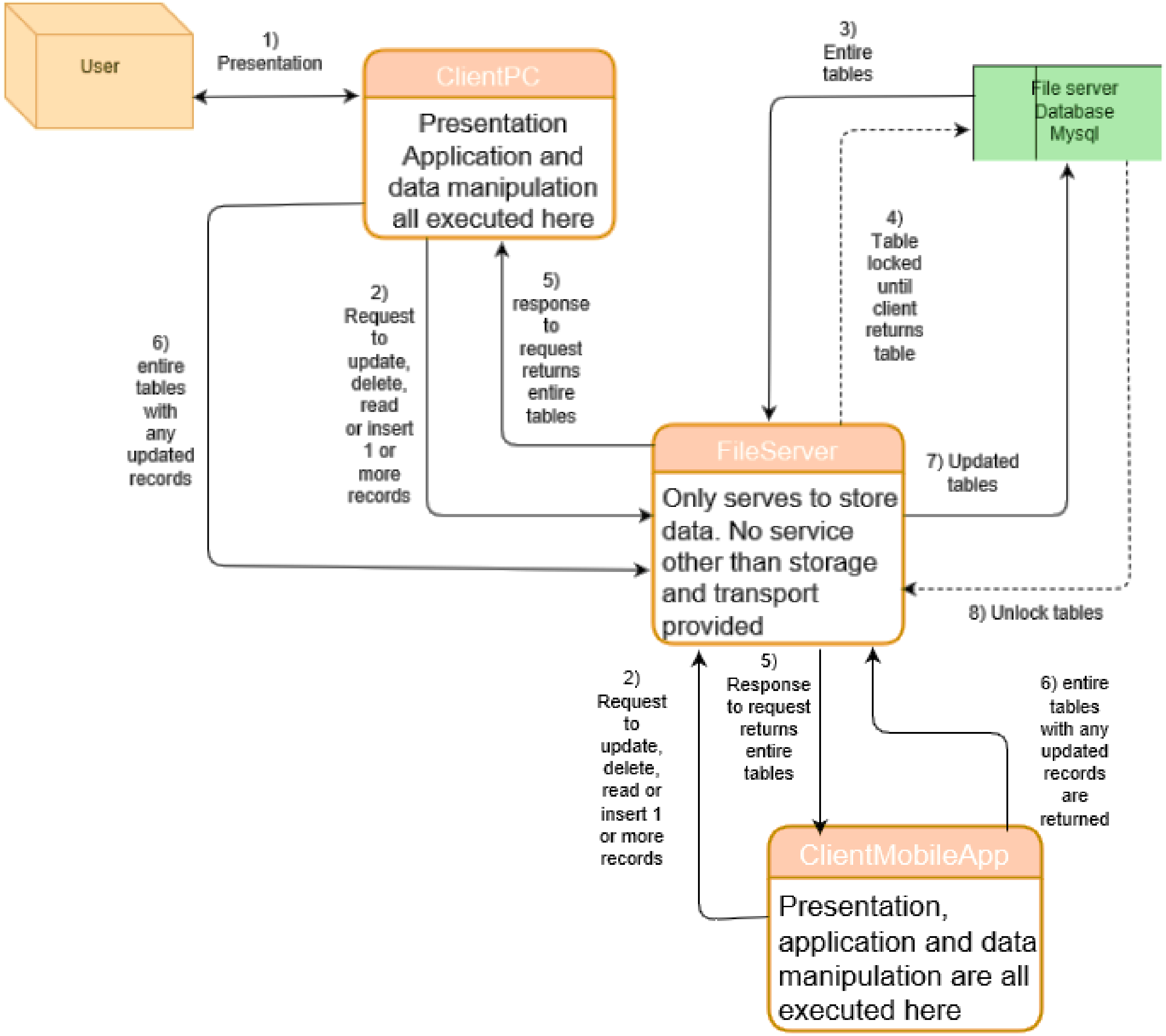
STRUCTURED DIAGRAM

The diagram below represents our system's structured diagram



FILE/SERVER ARCHITECTURE

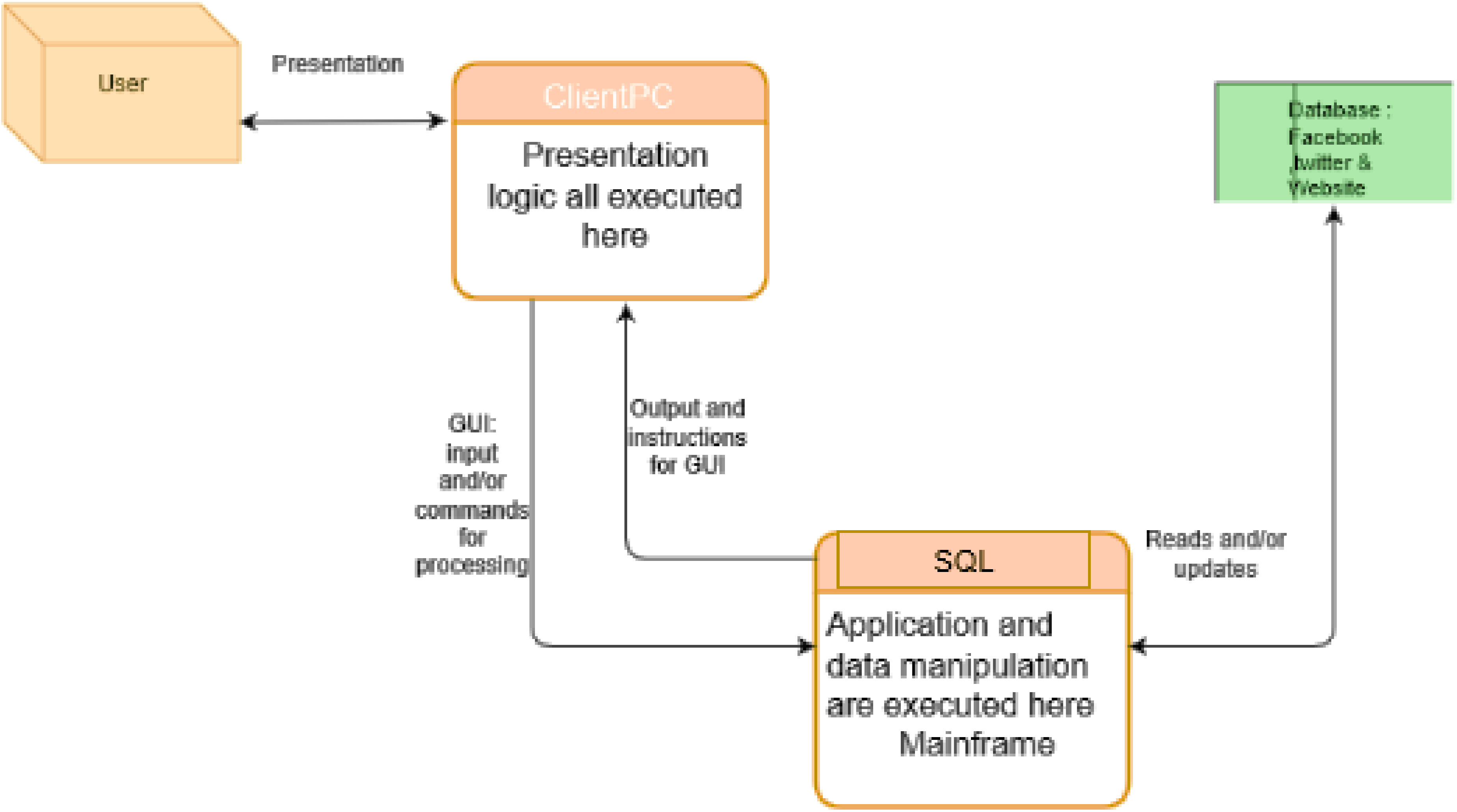
The diagram below file/server architecture of our system



User
1) Presentation
ClientPC
Presentation Application and data manipulation all executed here
3) Entire tables
File server Database Mysql
4) Table locked until client returns table
2) Request to update, delete, read or insert 1 or more records
5) response to request returns entire tables
6) entire tables with any updated records
FileServer
Only serves to store data. No service other than storage and transport provided
7) Updated tables
8) Unlock tables
2) Request to update, delete, read or insert 1 or more records
5) Response to request returns entire tables
6) entire tables with any updated records are returned
ClientMobileApp
Presentation, application and data manipulation are all executed here

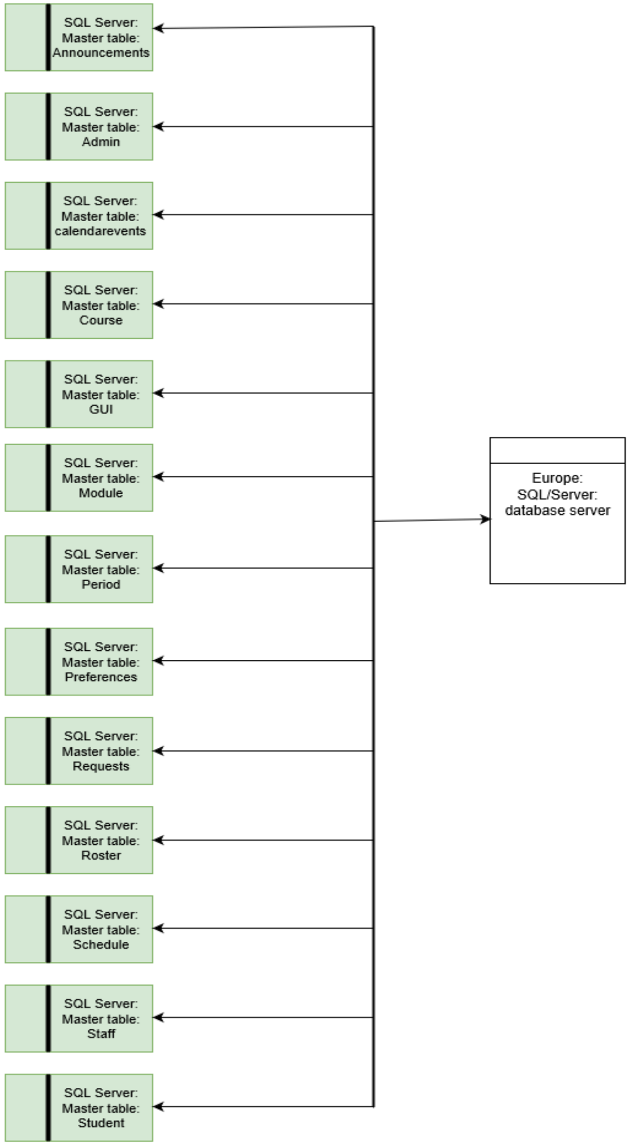
CLIENT/SERVER ARCHITECTURE

The diagram below client/server architecture of our system


User
Presentation
ClientPC
Presentation logic all executed here
Database : Facebook twitter & Website
GUI: input and/or commands for processing
Output and instructions for GUI
SQL
Reads and/or updates
Application and data manipulation are executed here Mainframe

DATA DISTRIBUTION DIAGRAM

The diagram below shows how our data is distributed in our database

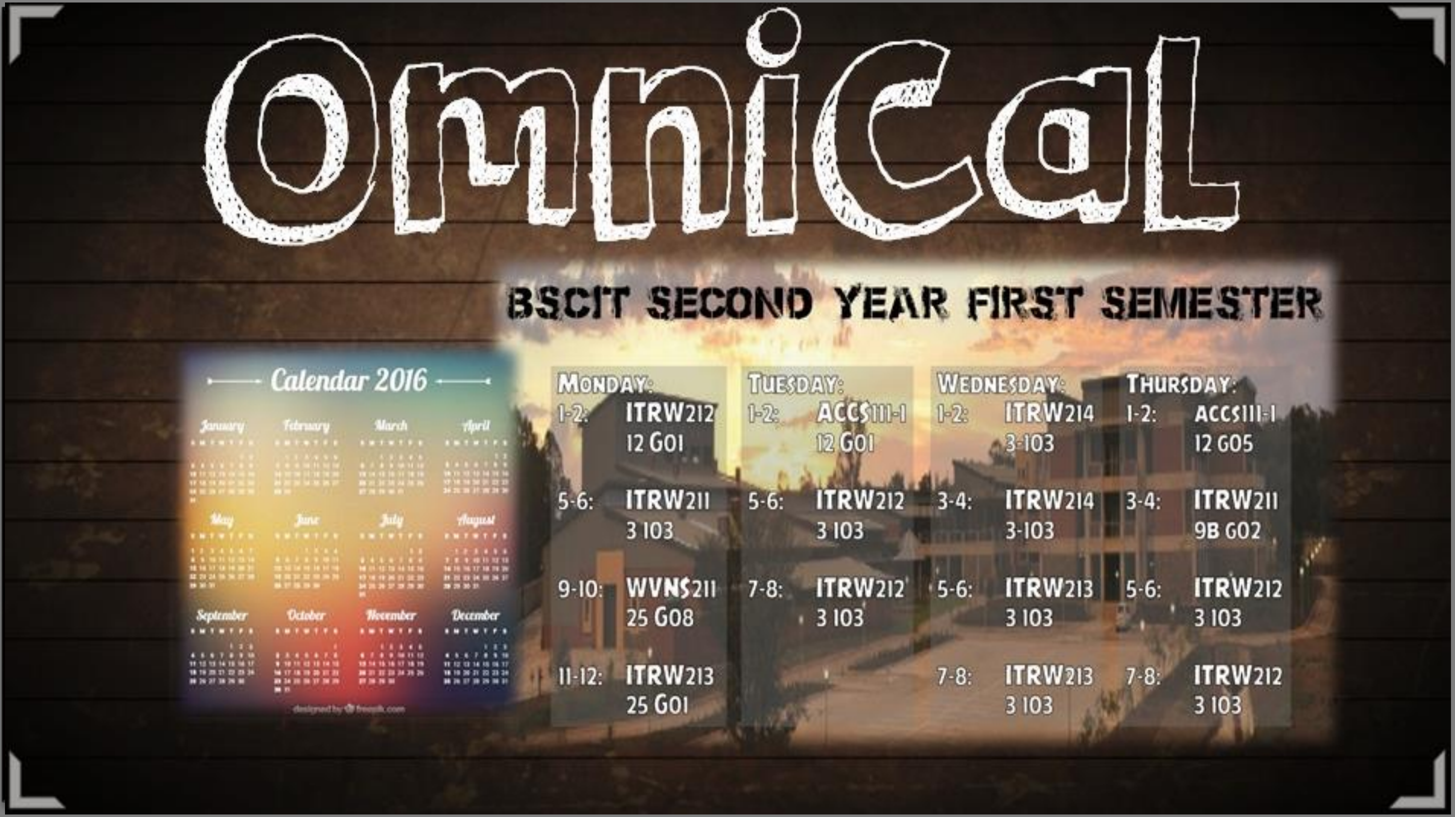


OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:	Tuesday:	Wednesday:	Thursday:
1-2: ITRW212 12 G01	1-2: ACCS111-1 12 G01	1-2: ITRW214 3-103	1-2: ACCS111-1 12 G05
5-6: ITRW211 3 103	5-6: ITRW212 3 103	3-4: ITRW214 3-103	3-4: ITRW211 9B G02
9-10: WVNS211 25 G08	7-8: ITRW212 3 103	5-6: ITRW213 3 103	5-6: ITRW212 3 103
11-12: ITRW213 25 G01		7-8: ITRW213 3-103	7-8: ITRW212 3 103

PROCESS MODELLING



OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

MONDAY:
1-2: ITRW212 12 G01
5-6: ITRW211 3 103
9-10: WVNS211 25 G08
11-12: ITRW213 25 G01

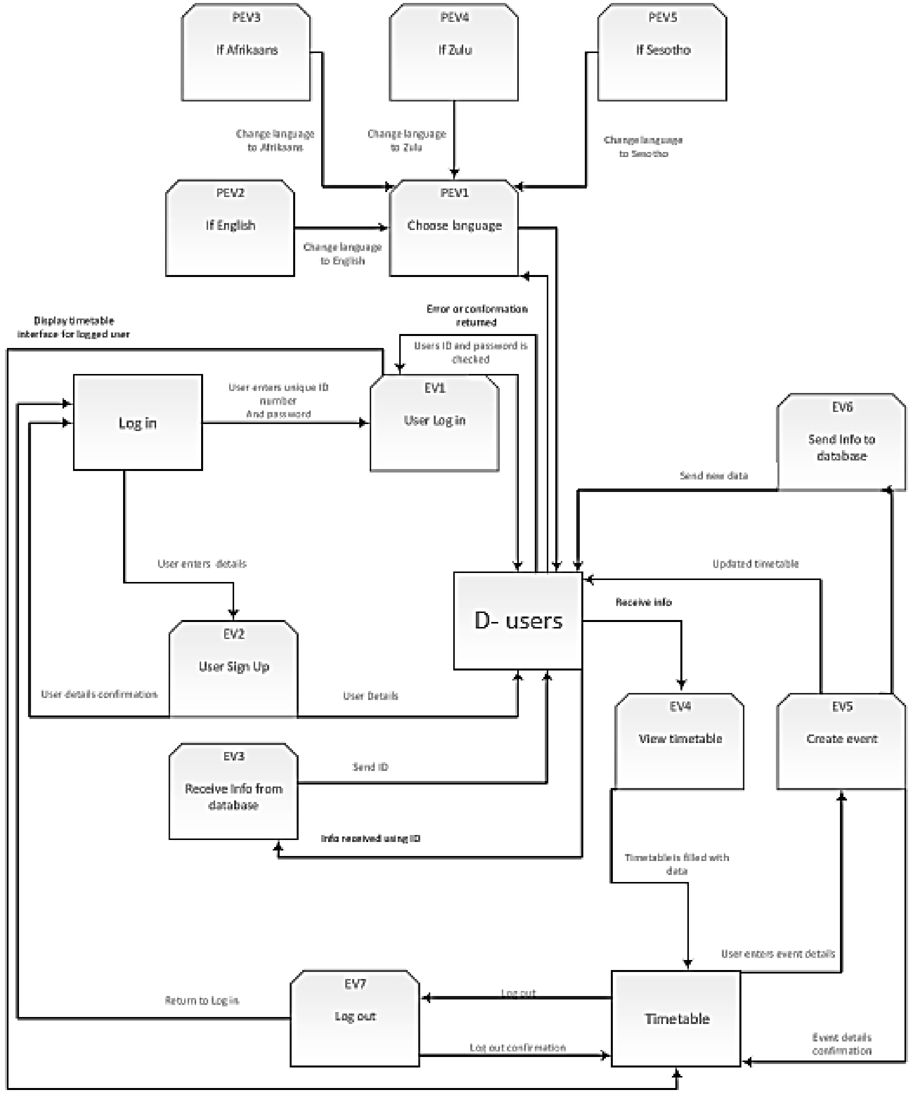
TUESDAY:
1-2: ACCS111-1 12 G01
5-6: ITRW212 3 103
7-8: ITRW212 3 103

WEDNESDAY:
1-2: ITRW214 3-103
3-4: ITRW214 3-103
5-6: ITRW213 3 103
7-8: ITRW213 3 103

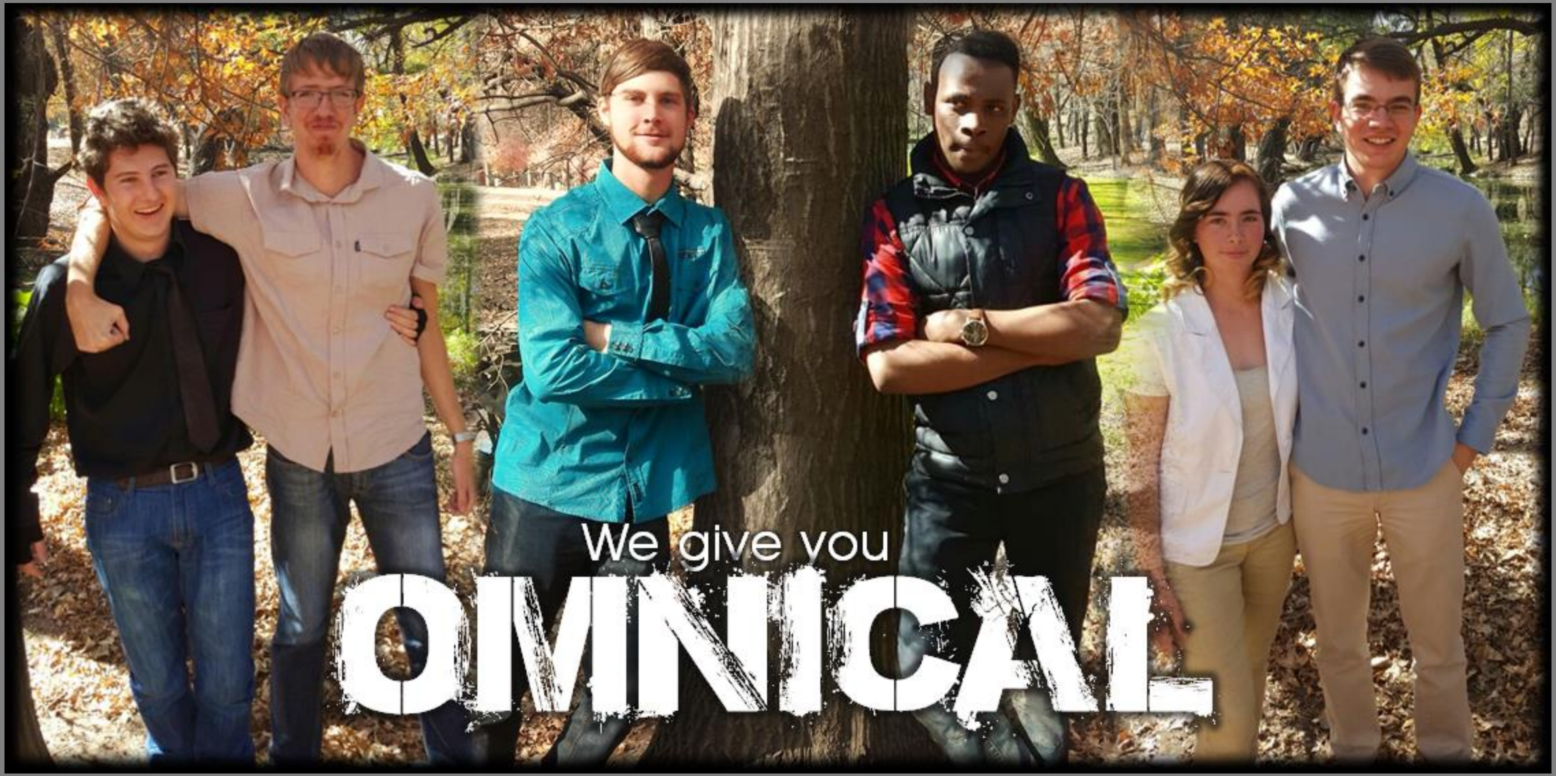
THURSDAY:
1-2: ACCS111-1 12 G05
3-4: ITRW211 9B G02
5-6: ITRW212 3 103
7-8: ITRW212 3 103

LOGICAL DATA FLOW DIAGRAM

OmniCal's Primitive Data Flow Diagram shows a more detailed flow of data between events than the Event Data Diagram . This is our logical data flow diagram

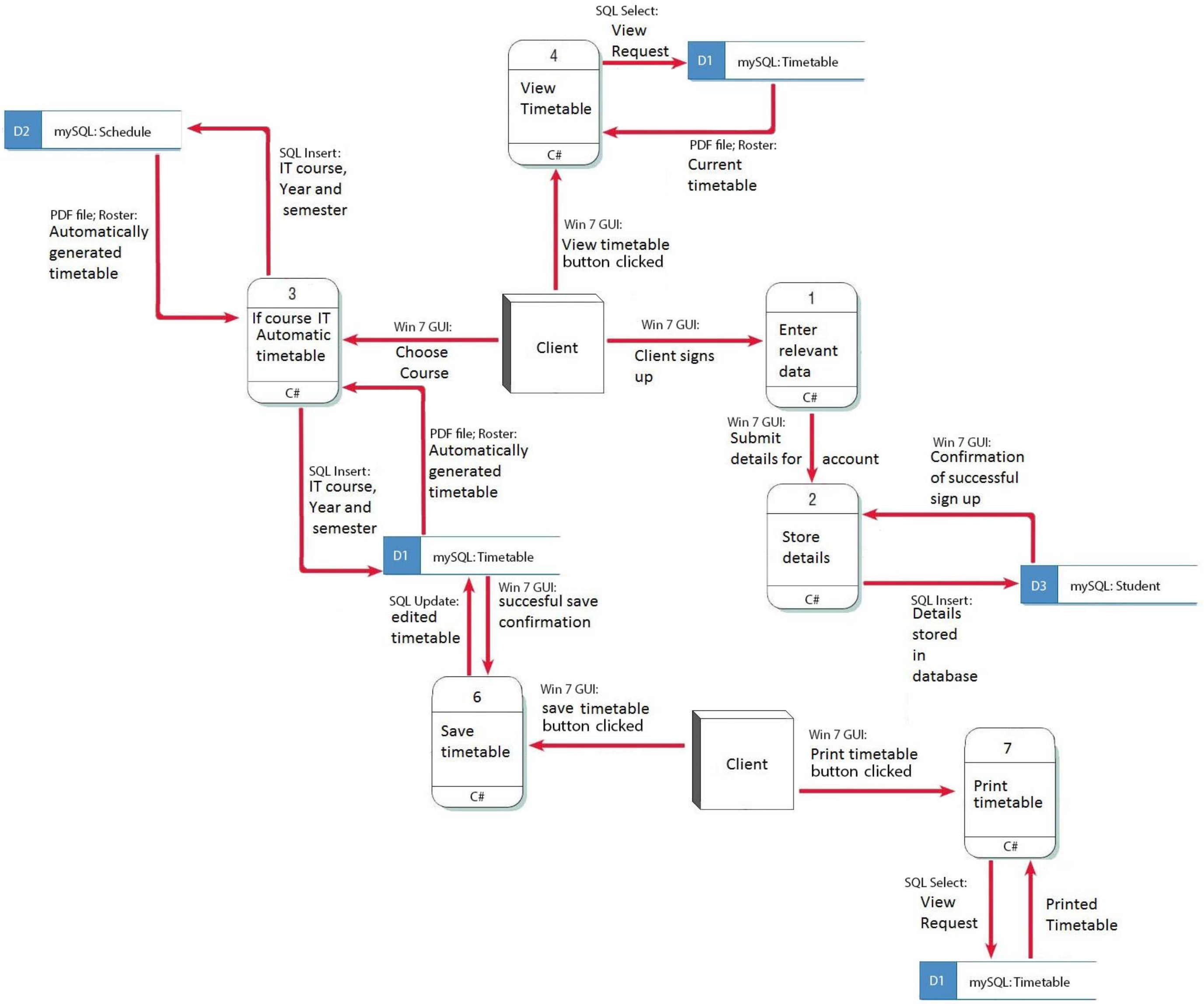


PHYSCIAL DFD



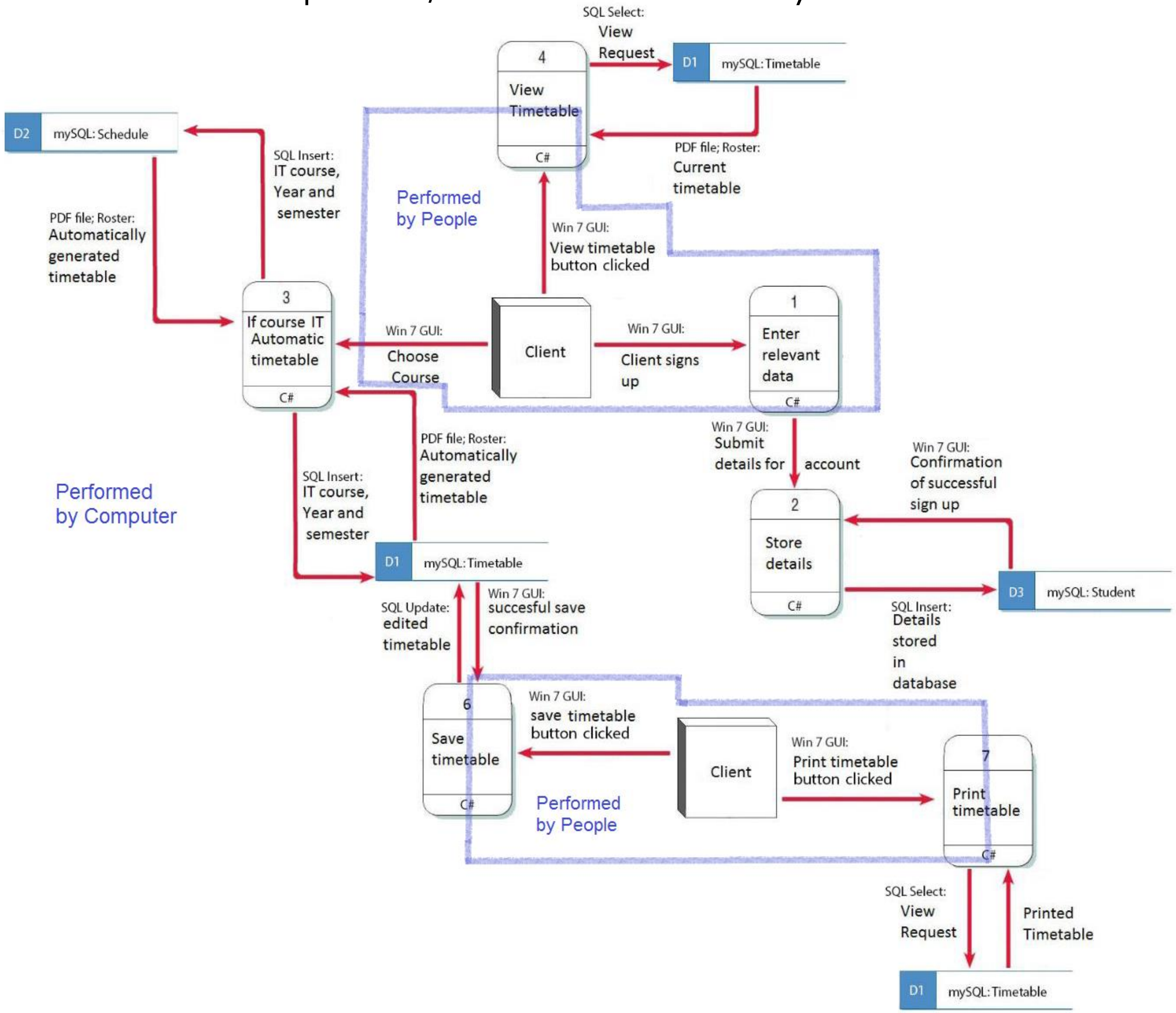
PHYSICAL DATA FLOW DIAGRAM

The following diagram represents OmniCal's physical data flow



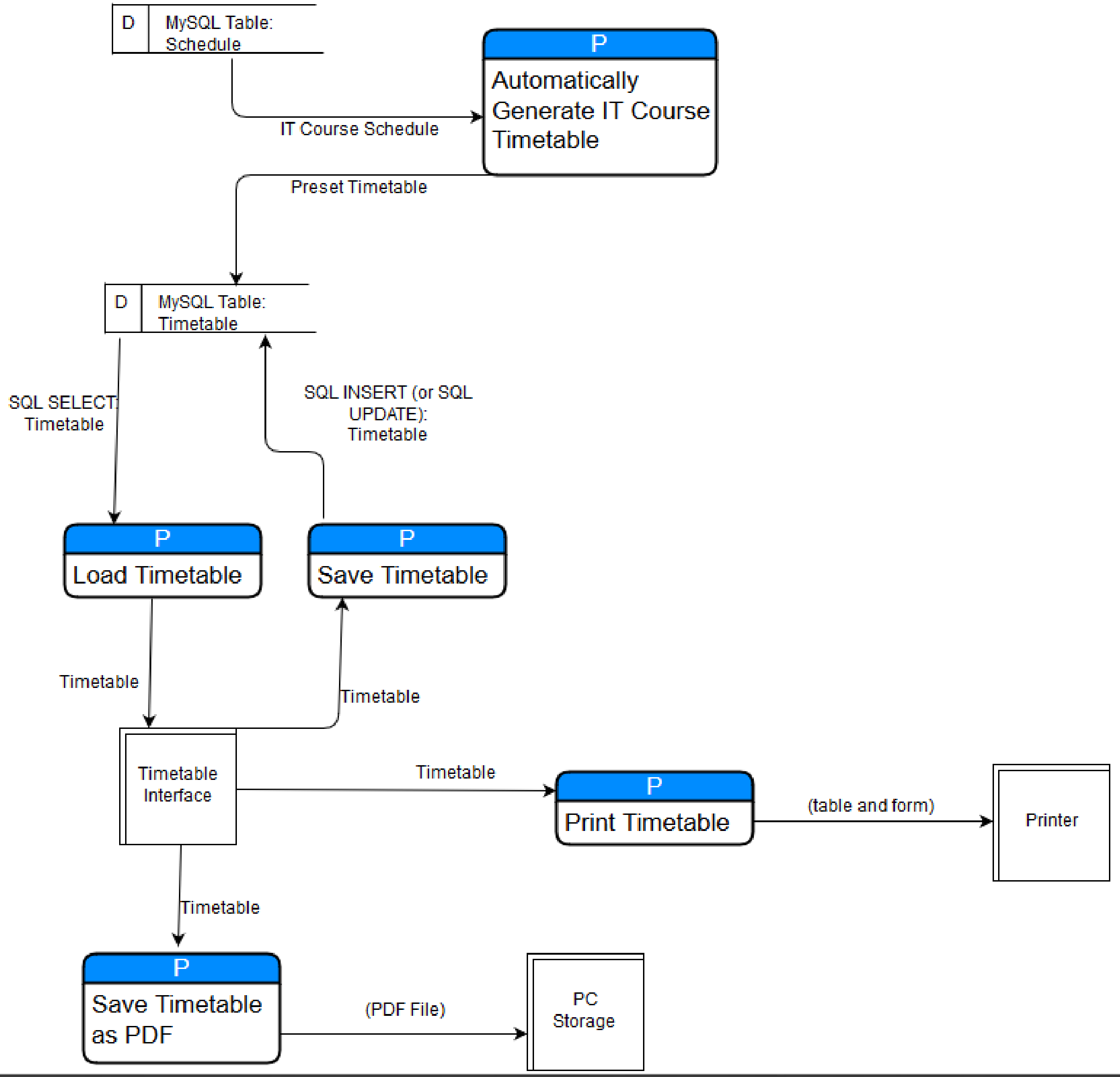
PERSON/ MACHINE BOUNDARY

The following diagram represents OmniCal's physical data flow with a person/ machine boundary line


SQL Select: View Request
4
View Timetable
C#
D1
mySQL: Timetable
PDF file; Roster: Current timetable
D2
mySQL: Schedule
SQL Insert: IT course, Year and semester
PDF file; Roster: Automatically generated timetable
Performed by People
Win 7 GUI: View timetable button clicked
3
If course IT Automatic timetable
C#
Win 7 GUI: Choose Course
Client
Win 7 GUI: Client signs up
1
Enter relevant data
C#
Win 7 GUI: Submit details for account
Win 7 GUI: Confirmation of successful sign up
PDF file; Roster: Automatically generated timetable
SQL Insert: IT course, Year and semester
Performed by Computer
2
Store details
C#
D1
mySQL: Timetable
D3
mySQL: Student
SQL Insert: Details stored in database
SQL Update: edited timetable
Win 7 GUI: succesful save confirmation
6
Save timetable
C#
Win 7 GUI: save timetable button clicked
Client
Win 7 GUI: Print timetable button clicked
7
Print timetable
C#
Performed by People
SQL Select: View Request
Printed Timetable
D1
mySQL: Timetable

MANUAL UNIT

The following diagram represents OmniCal's manual unit



OmniCal

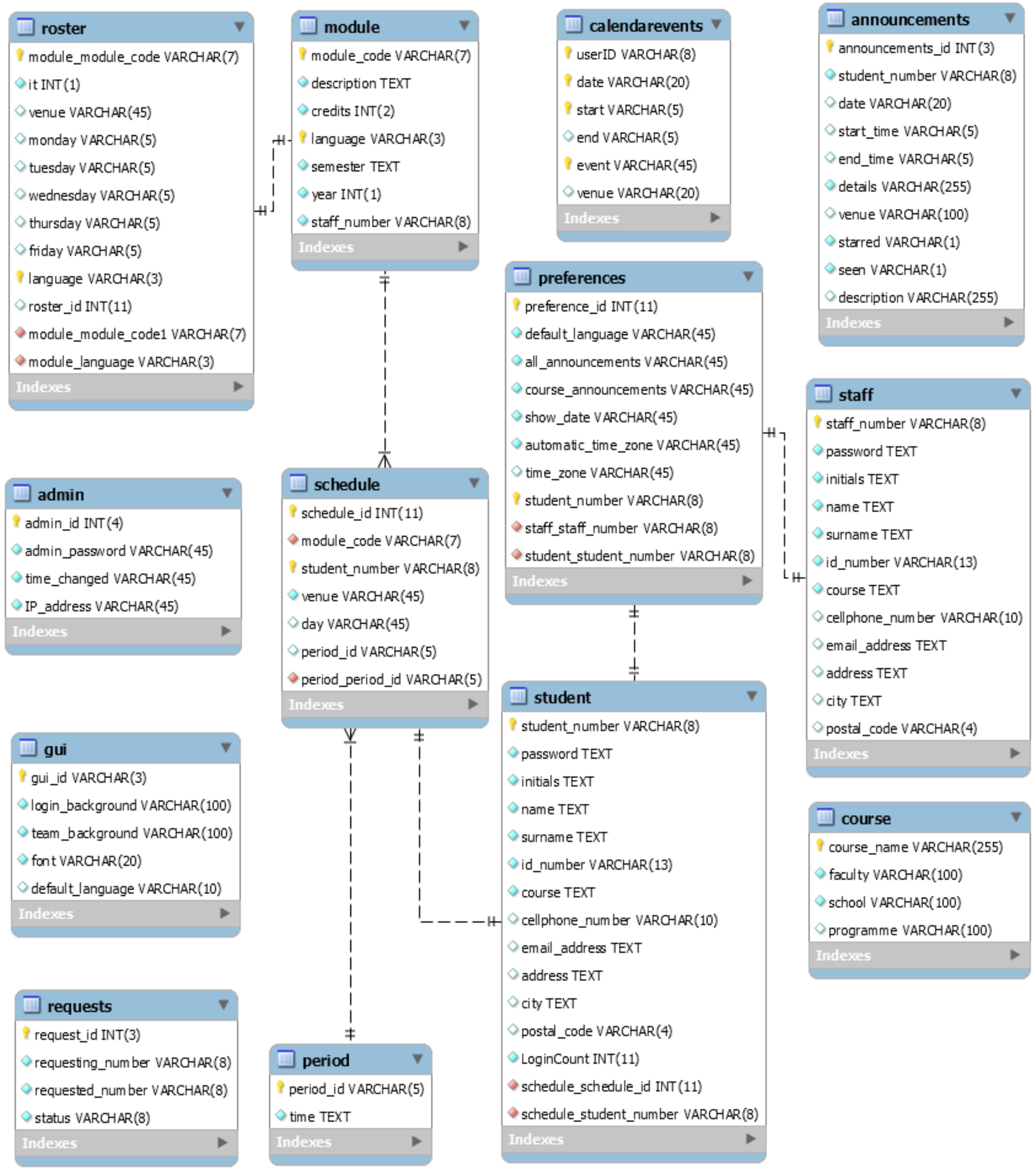
BSCIT SECOND YEAR FIRST SEMESTER

Monday:	Tuesday:	Wednesday:	Thursday:
1-2: ITRW212 12 G01	1-2: ACCS111-1 12 G01	1-2: ITRW214 3-103	1-2: ACCS111-1 12 G05
5-6: ITRW211 3 103	5-6: ITRW212 3 103	3-4: ITRW214 3-103	3-4: ITRW211 9B G02
9-10: WVNS211 25 G08	7-8: ITRW212 3 103	5-6: ITRW213 3 103	5-6: ITRW212 3 103
11-12: ITRW213 25 G01		7-8: ITRW213 3 103	7-8: ITRW212 3 103

LOGICAL DATA MODEL

LOGICAL DATA MODEL

OmniCal makes use of a MySQL database that contains 13 tables which communicate with the users as follows:



roster
module_module_code VARCHAR(7)
it INT(1)
venue VARCHAR(45)
monday VARCHAR(5)
tuesday VARCHAR(5)
wednesday VARCHAR(5)
thursday VARCHAR(5)
friday VARCHAR(5)
language VARCHAR(3)
roster_id INT(11)
module_module_code1 VARCHAR(7)
module_language VARCHAR(3)
Indexes

module
module_code VARCHAR(7)
description TEXT
credits INT(2)
language VARCHAR(3)
semester TEXT
year INT(1)
staff_number VARCHAR(8)
Indexes

calendarevents
userID VARCHAR(8)
date VARCHAR(20)
start VARCHAR(5)
end VARCHAR(5)
event VARCHAR(45)
venue VARCHAR(20)
Indexes

announcements
announcements_id INT(3)
student_number VARCHAR(8)
date VARCHAR(20)
start_time VARCHAR(5)
end_time VARCHAR(5)
details VARCHAR(255)
venue VARCHAR(100)
starred VARCHAR(1)
seen VARCHAR(1)
description VARCHAR(255)
Indexes

preferences
preference_id INT(11)
default_language VARCHAR(45)
all_announcements VARCHAR(45)
course_announcements VARCHAR(45)
show_date VARCHAR(45)
automatic_time_zone VARCHAR(45)
time_zone VARCHAR(45)
student_number VARCHAR(8)
staff_staff_number VARCHAR(8)
student_student_number VARCHAR(8)
Indexes

staff
staff_number VARCHAR(8)
password TEXT
initials TEXT
name TEXT
surname TEXT
id_number VARCHAR(13)
course TEXT
cellphone_number VARCHAR(10)
email_address TEXT
address TEXT
city TEXT
postal_code VARCHAR(4)
Indexes

admin
admin_id INT(4)
admin_password VARCHAR(45)
time_changed VARCHAR(45)
IP_address VARCHAR(45)
Indexes

schedule
schedule_id INT(11)
module_code VARCHAR(7)
student_number VARCHAR(8)
venue VARCHAR(45)
day VARCHAR(45)
period_id VARCHAR(5)
period_period_id VARCHAR(5)
Indexes

student
student_number VARCHAR(8)
password TEXT
initials TEXT
name TEXT
surname TEXT
id_number VARCHAR(13)
course TEXT
cellphone_number VARCHAR(10)
email_address TEXT
address TEXT
city TEXT
postal_code VARCHAR(4)
LoginCount INT(11)
schedule_schedule_id INT(11)
schedule_student_number VARCHAR(8)
Indexes

course
course_name VARCHAR(255)
faculty VARCHAR(100)
school VARCHAR(100)
programme VARCHAR(100)
Indexes

gui
gui_id VARCHAR(3)
login_background VARCHAR(100)
team_background VARCHAR(100)
font VARCHAR(20)
default_language VARCHAR(10)
Indexes

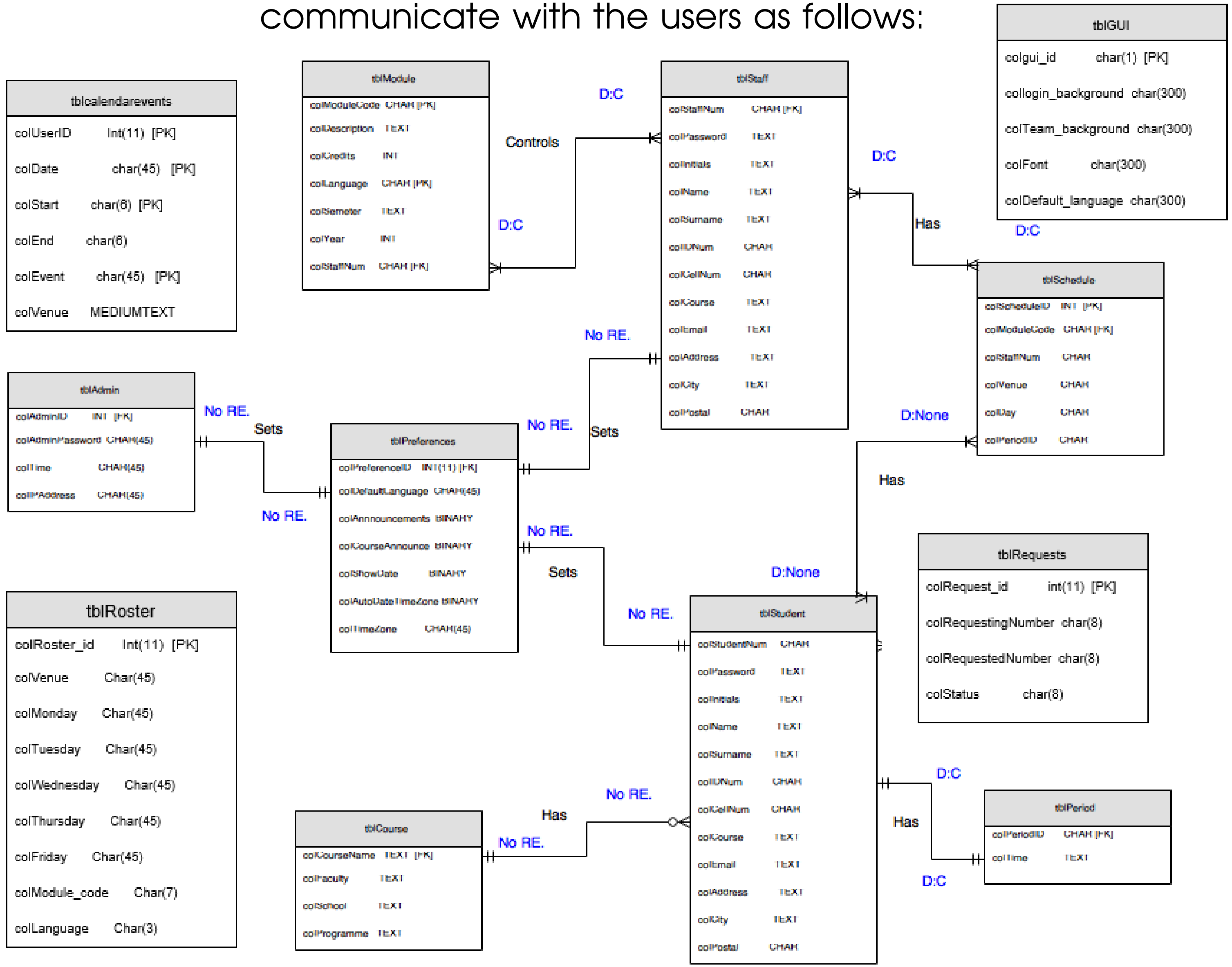
requests
request_id INT(3)
requesting_number VARCHAR(8)
requested_number VARCHAR(8)
status VARCHAR(8)
Indexes

period
period_id VARCHAR(5)
time TEXT
Indexes

DATABASE SCHEMA

DATABASE SCHEMA

OmniCal makes use of a MySQL database that contains 13 tables which communicate with the users as follows:



tblcalendarevents	
colUserID	Int(11) [PK]
colDate	char(45) [PK]
colStart	char(6) [PK]
colEnd	char(6)
colEvent	char(45) [PK]
colVenue	MEDIUMTEXT

tblModule	
colModuleCode	CHAR [PK]
colDescription	TEXT
colCredits	INT
colLanguage	CHAR [PK]
colSemeter	TEXT
colYear	INT
colStaffNum	CHAR [FK]

tblStaff	
colStaffNum	CHAR [PK]
colPassword	TEXT
colInitials	TEXT
colName	TEXT
colSurname	TEXT
colIDNum	CHAR
colCellNum	CHAR
colCourse	TEXT
colEmail	TEXT
colAddress	TEXT
colCity	TEXT
colPostal	CHAR

tblGUI	
colgui_id	char(1) [PK]
collogin_background	char(300)
colTeam_background	char(300)
colFont	char(300)
colDefault_language	char(300)

tblSchedule	
colScheduleID	INT [PK]
colModuleCode	CHAR [FK]
colStaffNum	CHAR
colVenue	CHAR
colDay	CHAR
colPeriodID	CHAR

tblAdmin	
colAdminID	INT [PK]
colAdminPassword	CHAR(45)
colTime	CHAR(45)
colIPAddress	CHAR(45)

tblPreferences	
colPreferenceID	INT(11) [PK]
colDefaultLanguage	CHAR(45)
colAnnouncements	BINARY
colCourseAnnounce	BINARY
colShowDate	BINARY
colAutoDateTimeZone	BINARY
colTimeZone	CHAR(45)

tblRequests	
colRequest_id	int(11) [PK]
colRequestingNumber	char(8)
colRequestedNumber	char(8)
colStatus	char(8)

tblRoster	
colRoster_id	Int(11) [PK]
colVenue	Char(45)
colMonday	Char(45)
colTuesday	Char(45)
colWednesday	Char(45)
colThursday	Char(45)
colFriday	Char(45)
colModule_code	Char(7)
colLanguage	Char(3)

tblStudent	
colStudentNum	CHAR
colPassword	TEXT
colInitials	TEXT
colName	TEXT
colSurname	TEXT
colIDNum	CHAR
colCellNum	CHAR
colCourse	TEXT
colEmail	TEXT
colAddress	TEXT
colCity	TEXT
colPostal	CHAR

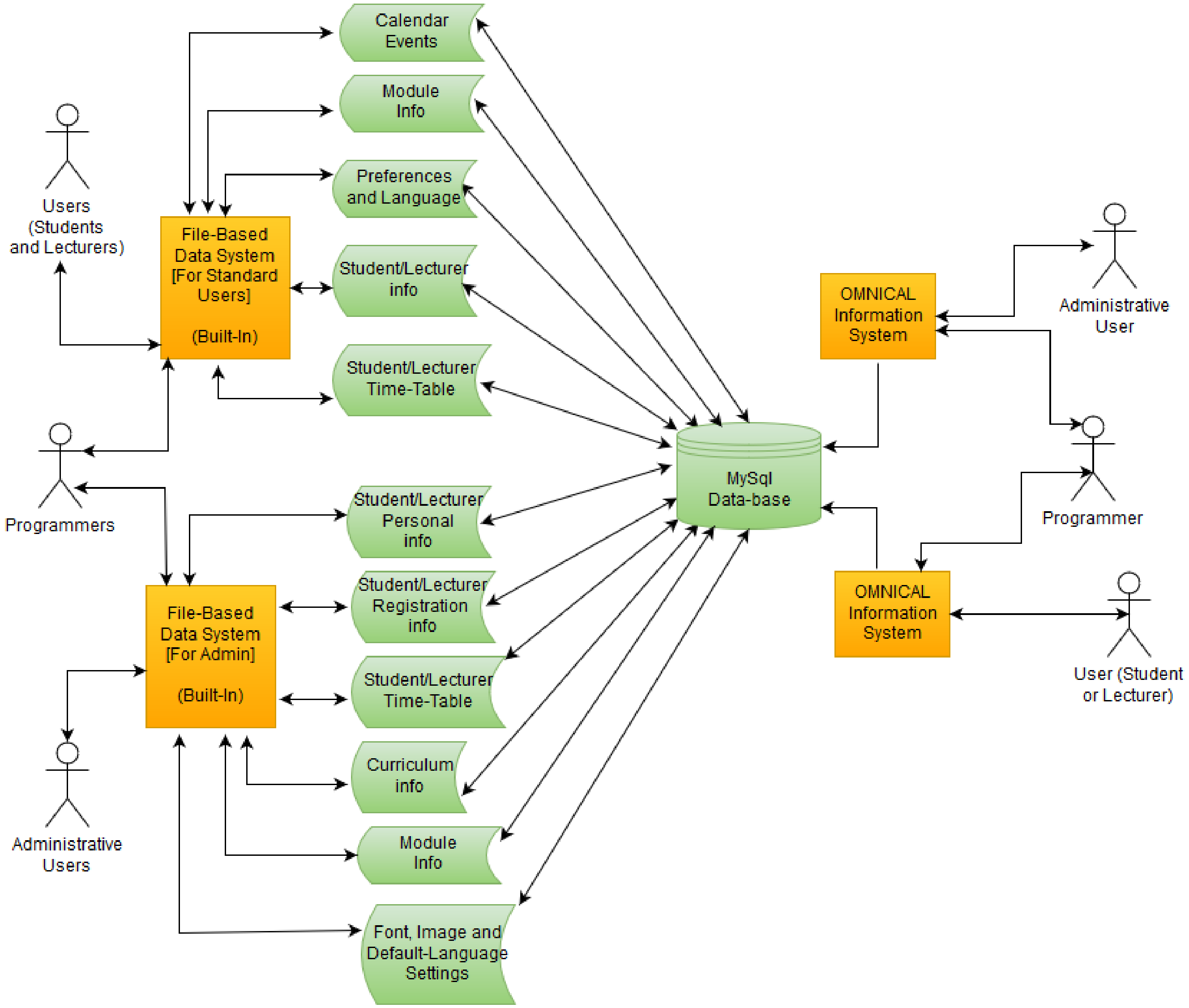
tblCourse	
colCourseName	TEXT [PK]
colFaculty	TEXT
colSchool	TEXT
colProgramme	TEXT

tblPeriod	
colPeriodID	CHAR [PK]
colTime	TEXT

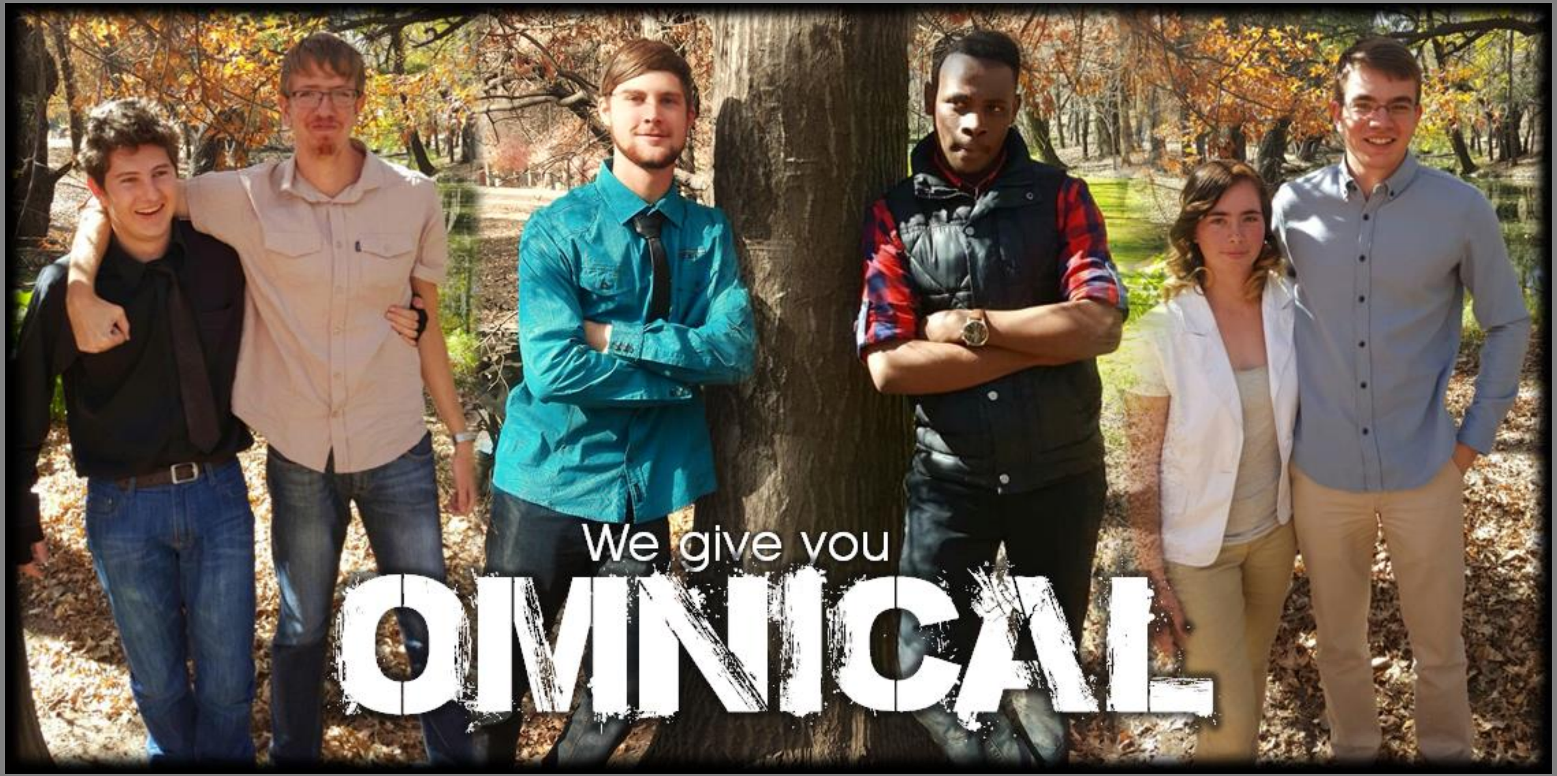
DATABASE ARCHITECTURE

DATABASE ARCHITECTURE

OmniCal makes use of a MySQL database that contains 13 tables which communicate with the users as follows:


Calendar Events
Module Info
Preferences and Language
Student/Lecturer info
Student/Lecturer Time-Table
Users (Students and Lecturers)
File-Based Data System [For Standard Users] (Built-In)
Programmers
Student/Lecturer Personal info
Student/Lecturer Registration info
Student/Lecturer Time-Table
Curriculum info
Module Info
Font, Image and Default-Language Settings
File-Based Data System [For Admin] (Built-In)
Administrative Users
MySql Data-base
OMNICAL Information System
Administrative User
Programmer
OMNICAL Information System
User (Student or Lecturer)

Capacity Planning



DATABASE CAPACITY PLANNING

The field sizes were calculated by adding the maximum length of each field for each table.

STEP 1 - Sum the field sizes:

Total characters: 3 140

- module = 181
- staff = 465
- preferences = 105
- admin = 139
- course = 555
- student = 465
- schedule = 121
- period = 20
- roster = 95
- announcements = 655
- calendarevents = 78
- gui = 233
- Requests = 27

DATABASE CAPACITY PLANNING (Cont)

The record sizes were obtained as the field sizes in Step 1 so that the following deductions can be made:

STEP 2 – Record size x entity instances (using growth over 3 years):

Growth = 1,4 x 1,4 x 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

STEP 3 – Sum the table sizes:

Total = 22 261 415,38

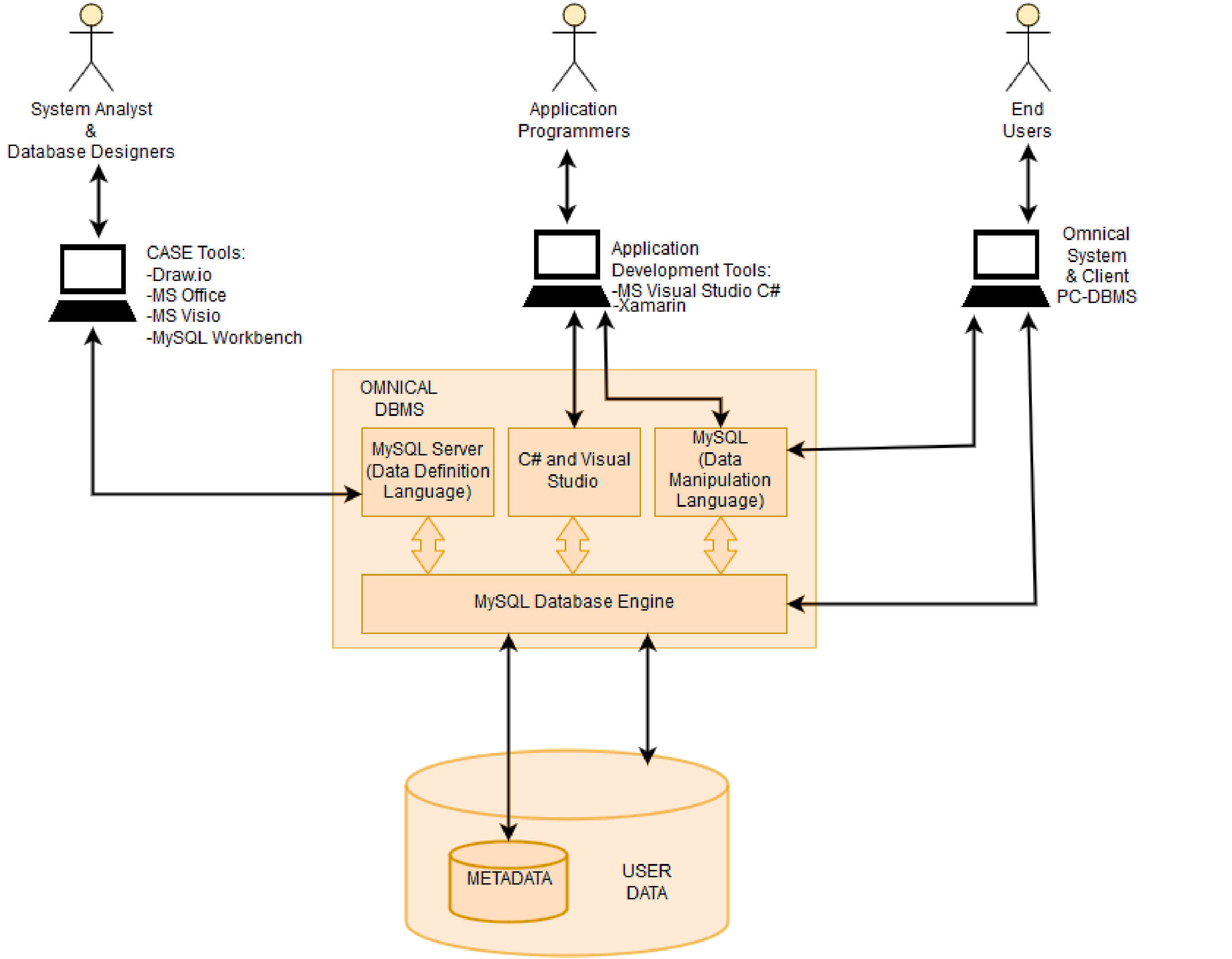
STEP 4 – Add slack capacity buffer (10%):

22 261 415,38 * 0.10 = 2 226 141,54

Anticipated database capacity: <u>2,3 GB</u>

DATABASE MANAGEMENT SYSTEM ARCHITECTURE (DBMS)

OmniCal's database management is illustrated below:



OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2:	ITRW212 12 G01	1-2:	ACCS111-1 12 G01	1-2:	ITRW214 3-103	1-2:	ACCS111-1 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3-103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3-103	7-8:	ITRW212 3 103

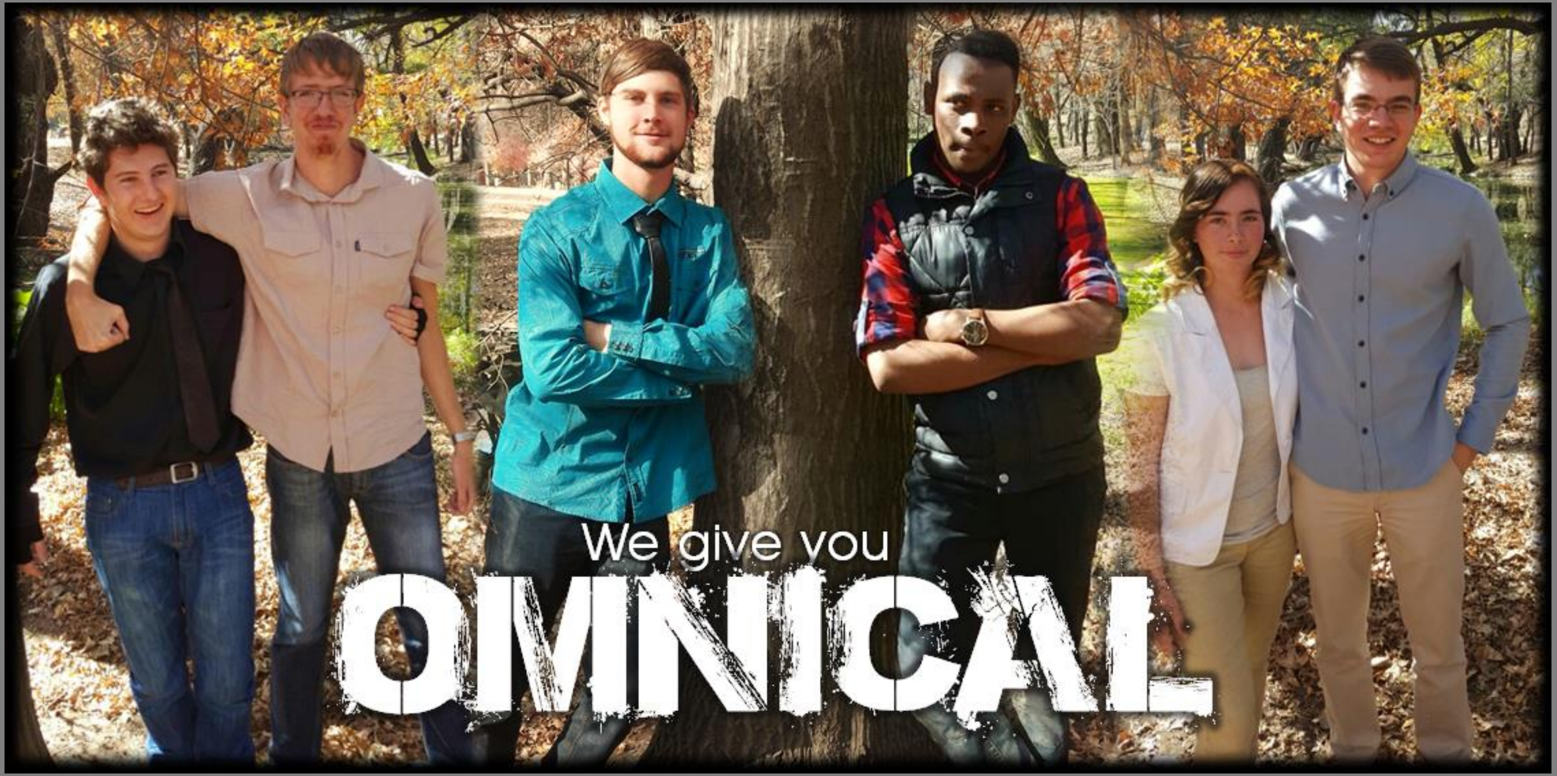
OUTPUT TAXONOMY

TAXONOMY FOR COMPUTER GENERATED OUTPUTS

OmniCal's computer generated outputs are laid out in the table below:

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

OUTPUT DESIGN



OUTPUT DESIGN GUIDELINES

OmniCal has followed and applied the following output guidelines:

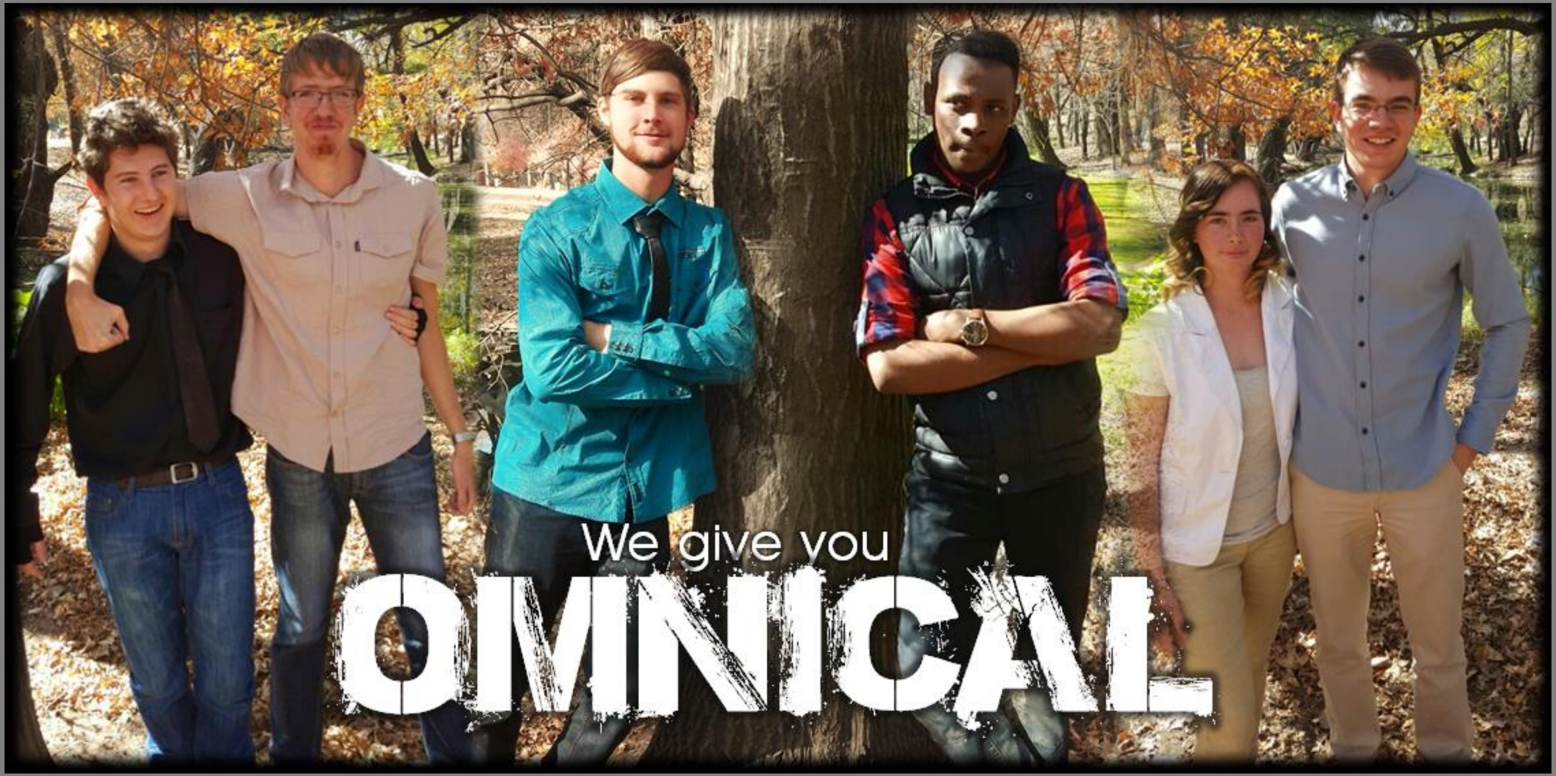
Guidelines	Applied
Simple to read and interpret	✓
Title for every output	✓
Time stamp every output	-
Reports and screens should include sections and headings to segment information	✓
Form base output - clearly labelled fields	✓
Tabular outputs - clearly ladled columns	✓
Reports should include legends 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	✓

OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2.	ITRW212 12 G01	1-2:	ACCS111-1 12 G01	1-2:	ITRW214 3-103	1-2:	ACCS111-1 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3-103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3-103	7-8:	ITRW212 3 103

INPUT TAXONOMY



TAXONOMY FOR COMPUTER GENERATED INPUTS

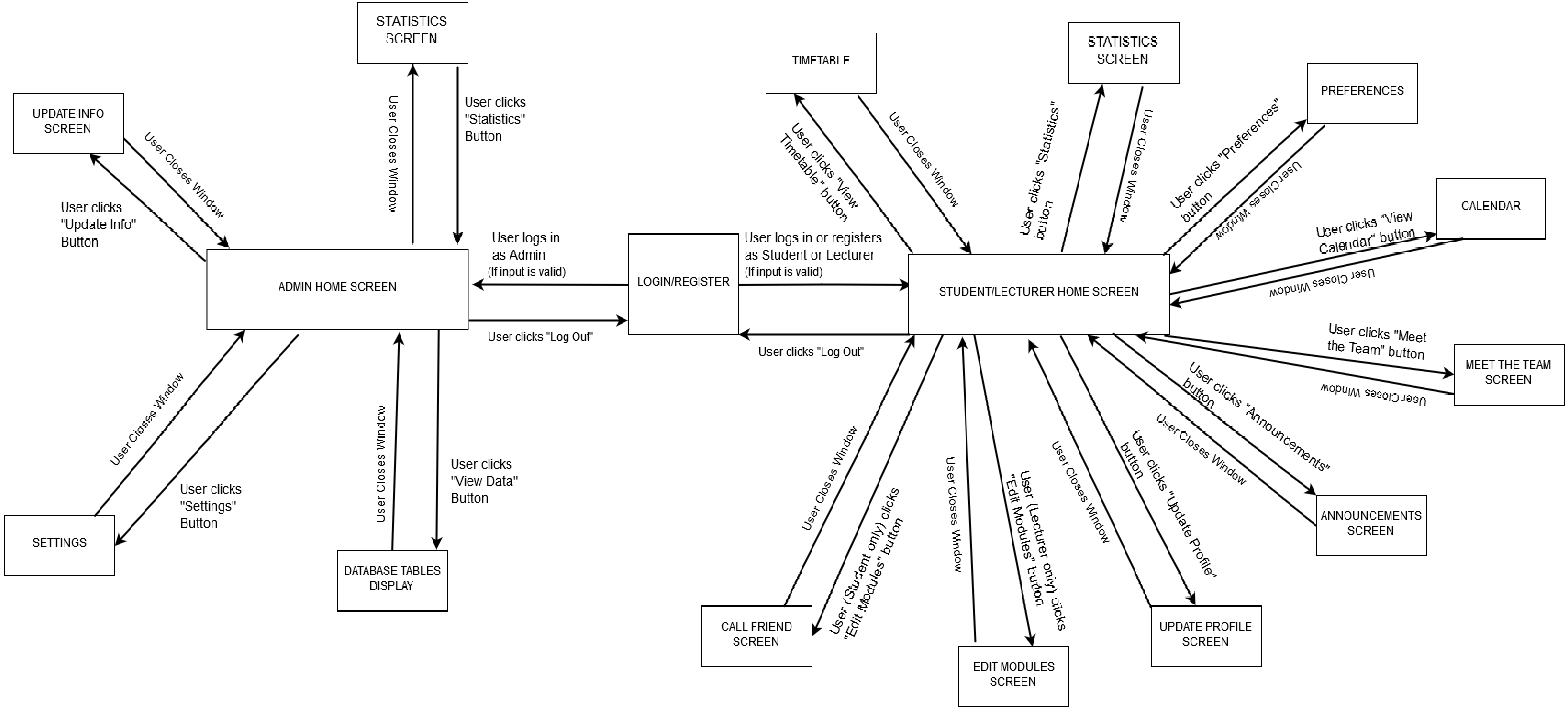
OmniCal's computer generated inputs are laid out in the table below:

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	- 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	- 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

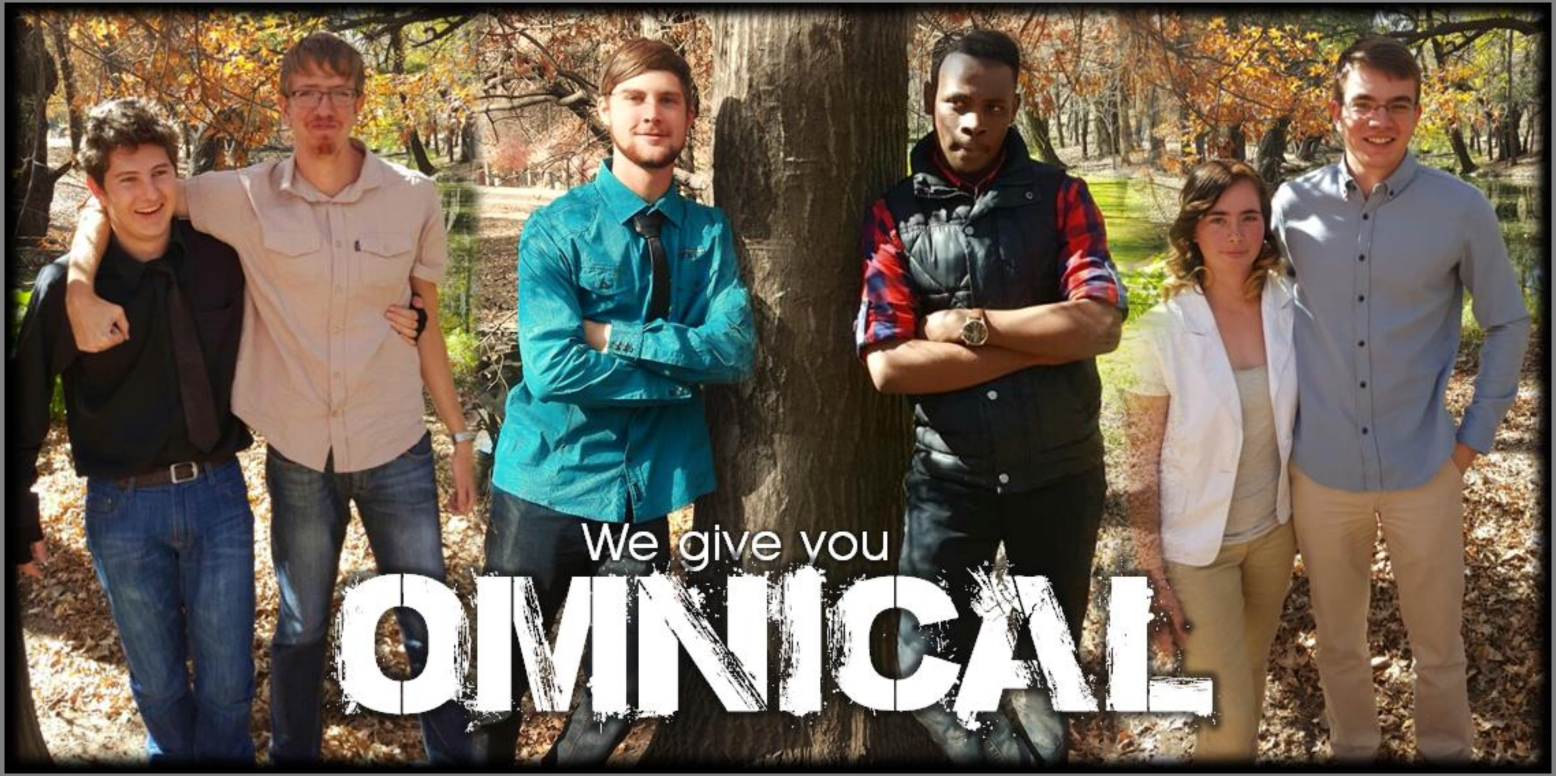
TRANSITION DIAGRAM

TRANSITION DIAGRAM

OmniCal's transition diagram is laid out below:



INPUT DESIGN



INPUT DESIGN GUIDELINES

OmniCal has followed and applied the following input guidelines:

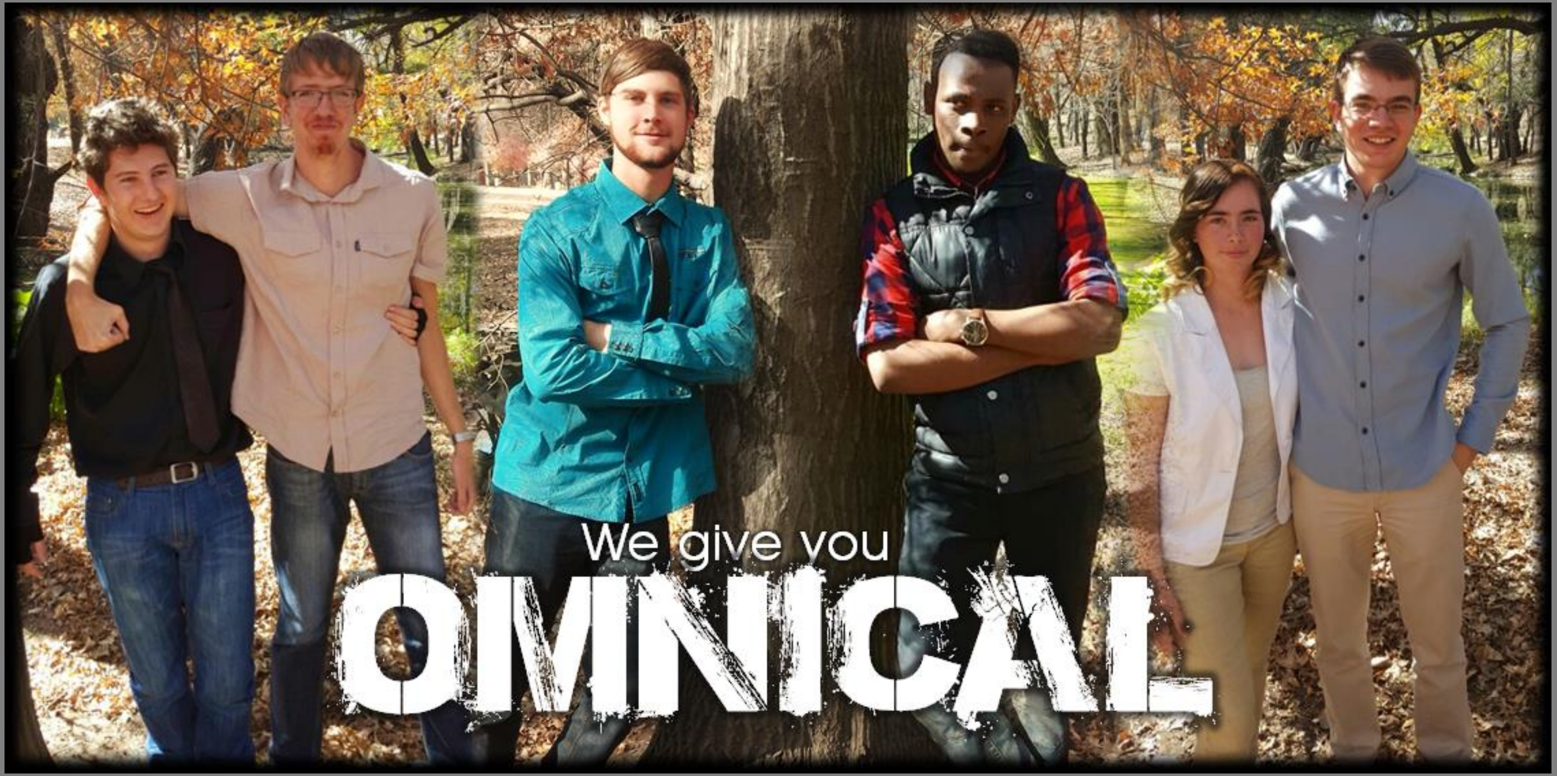
Guidelines	Applied
Capture only variable data	✓
Do not capture data that can be calculated or stored in computer programs	✓
Use codes for appropriate attributes	✓
Include instruction to complete forms	✓
Minimize handwriting/typing	✓
Sequenced data entry (Left to right and top to bottom)	✓
Use design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

OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2.	**ITRW**212 12 G01	1-2:	**ACCS**111-1 12 G01	1-2:	**ITRW**214 3-103	1-2:	**ACCS**111-1 12 G05
5-6:	**ITRW**211 3 103	5-6:	**ITRW**212 3 103	3-4:	**ITRW**214 3-103	3-4:	**ITRW**211 9B G02
9-10:	**WVNS**211 25 G08	7-8:	**ITRW**212 3 103	5-6:	**ITRW**213 3 103	5-6:	**ITRW**212 3 103
11-12:	**ITRW**213 25 G01			7-8:	**ITRW**213 3 103	7-8:	**ITRW**212 3 103

DIALOG CHARTS



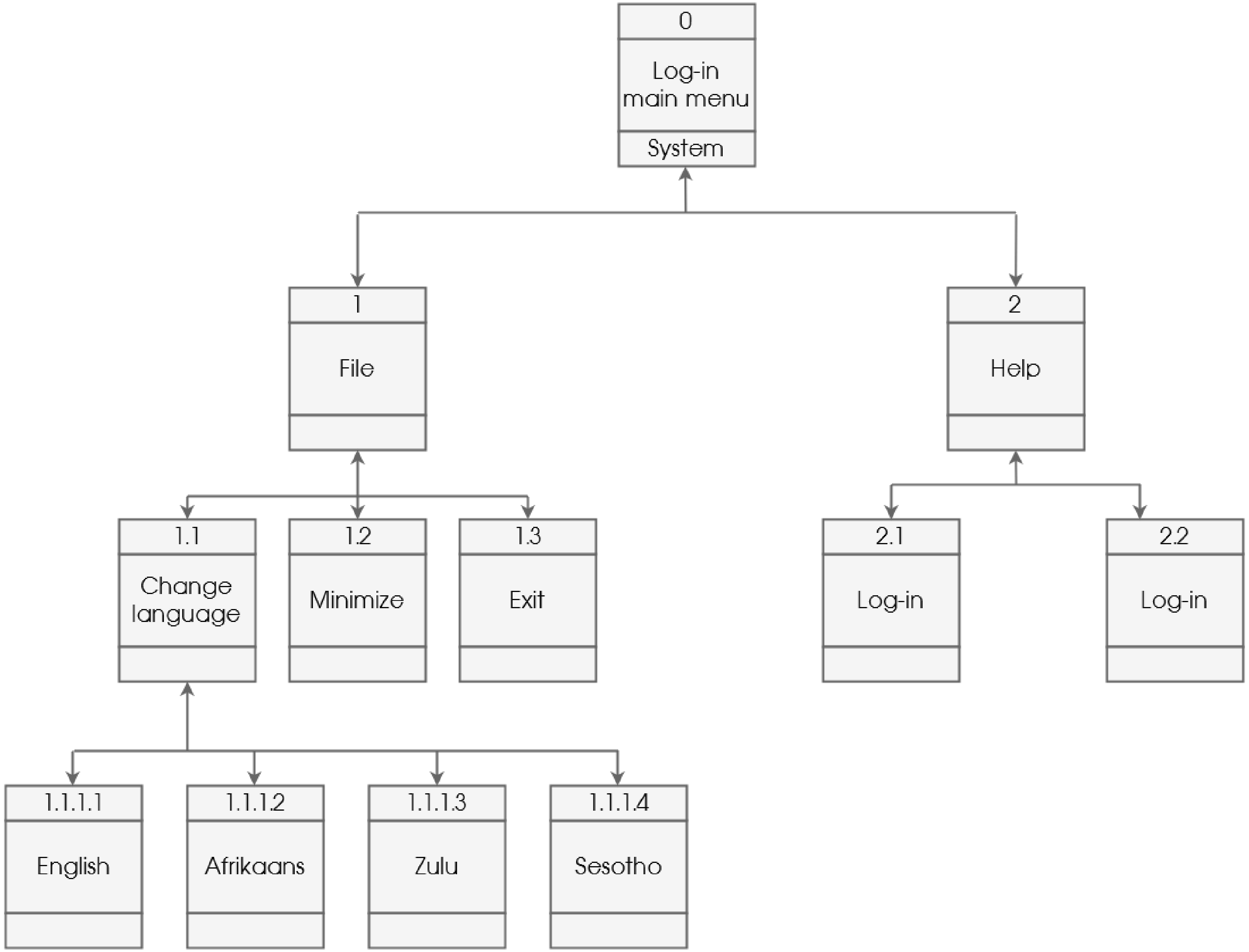
DIALOG CHART LIST

OmniCal has 5 dialog charts split between 10 forms as indicated below:

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	

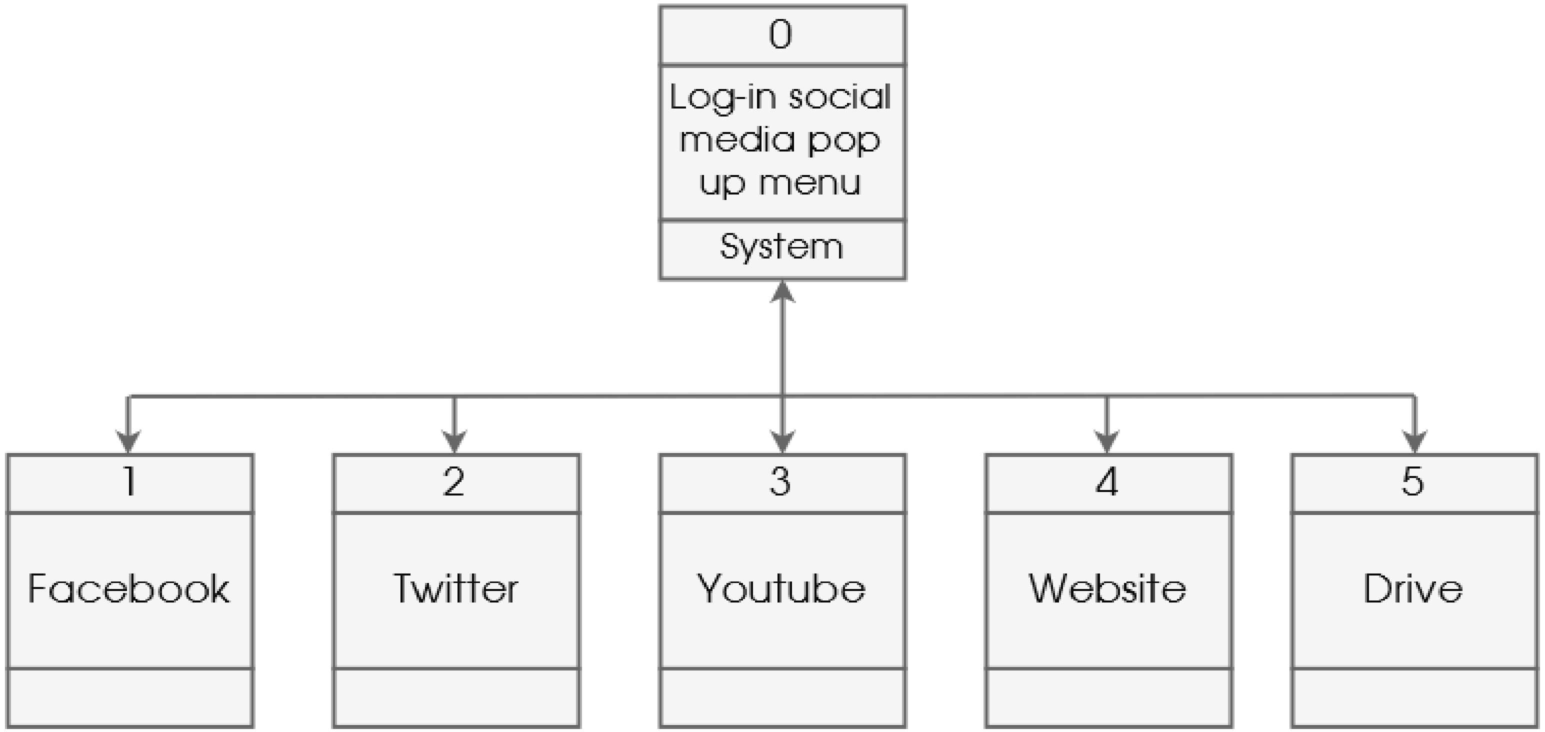
LOGIN MAIN MENU

OmniCal's login form has the following main menu layout:



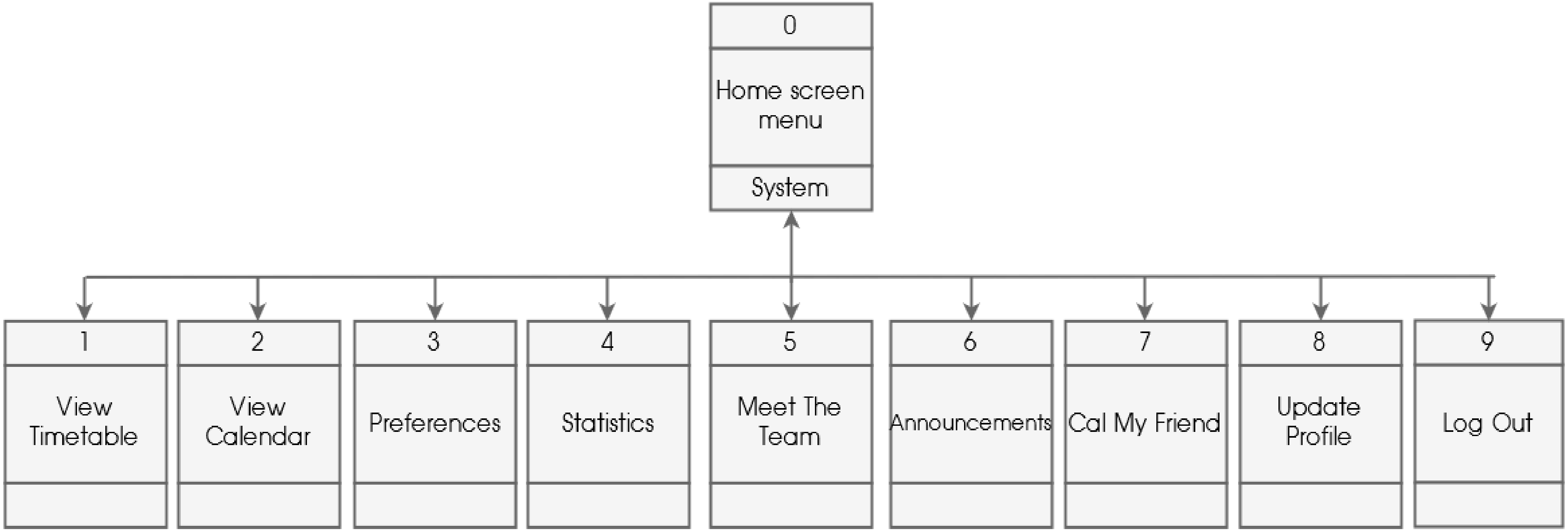
LOGIN SOCIAL MEDIA POP UP MENU

OmniCal's login form has the following social media pop up menu layout:



STUDENT HOME SCREEN MENU

A student using OmniCal would see the following menu on the home screen:



0
Home screen menu
System

1
View Timetable

2
View Calendar

3
Preferences

4
Statistics

5
Meet The Team

6
Announcements

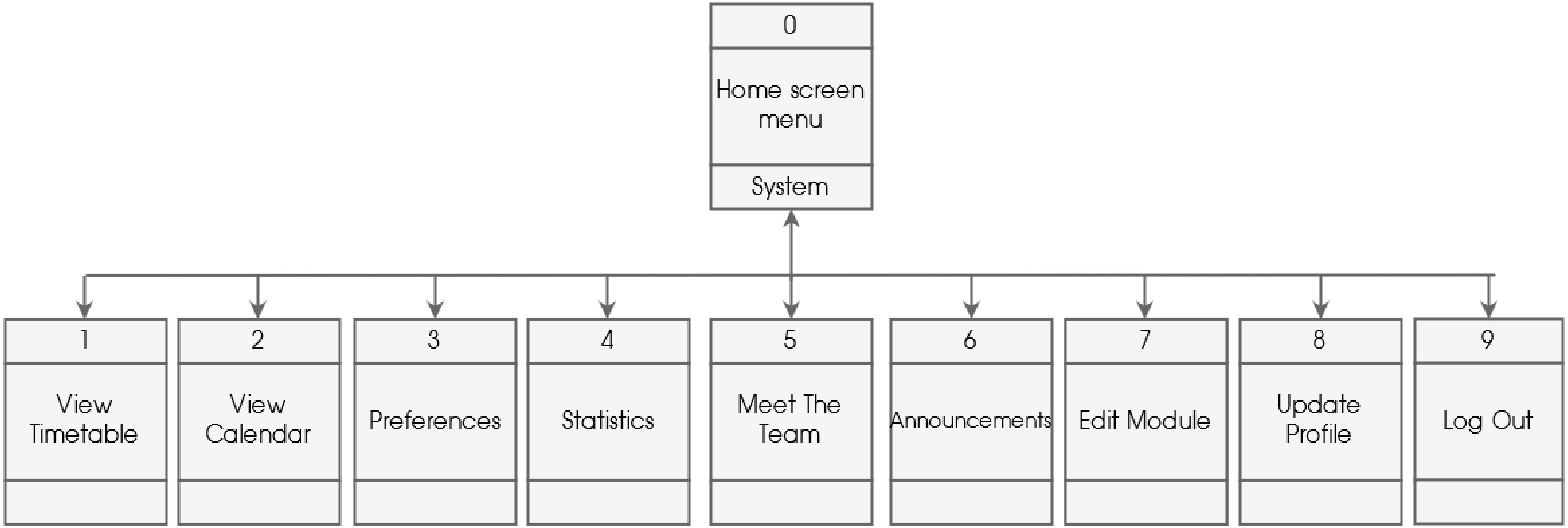
7
Cal My Friend

8
Update Profile

9
Log Out

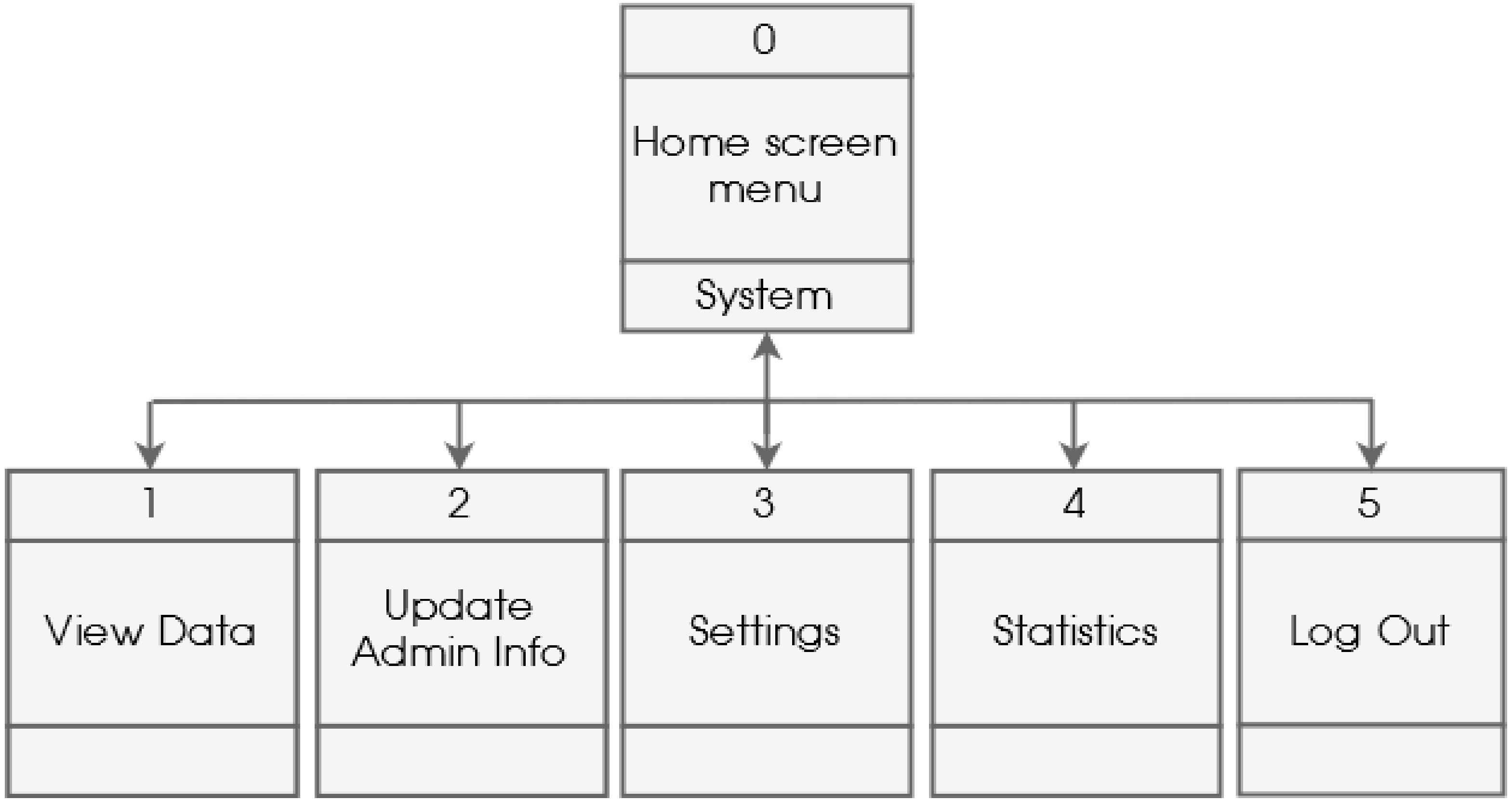
STAFF HOME SCREEN MENU

A staff member using OmniCal would see the following menu on the home screen:



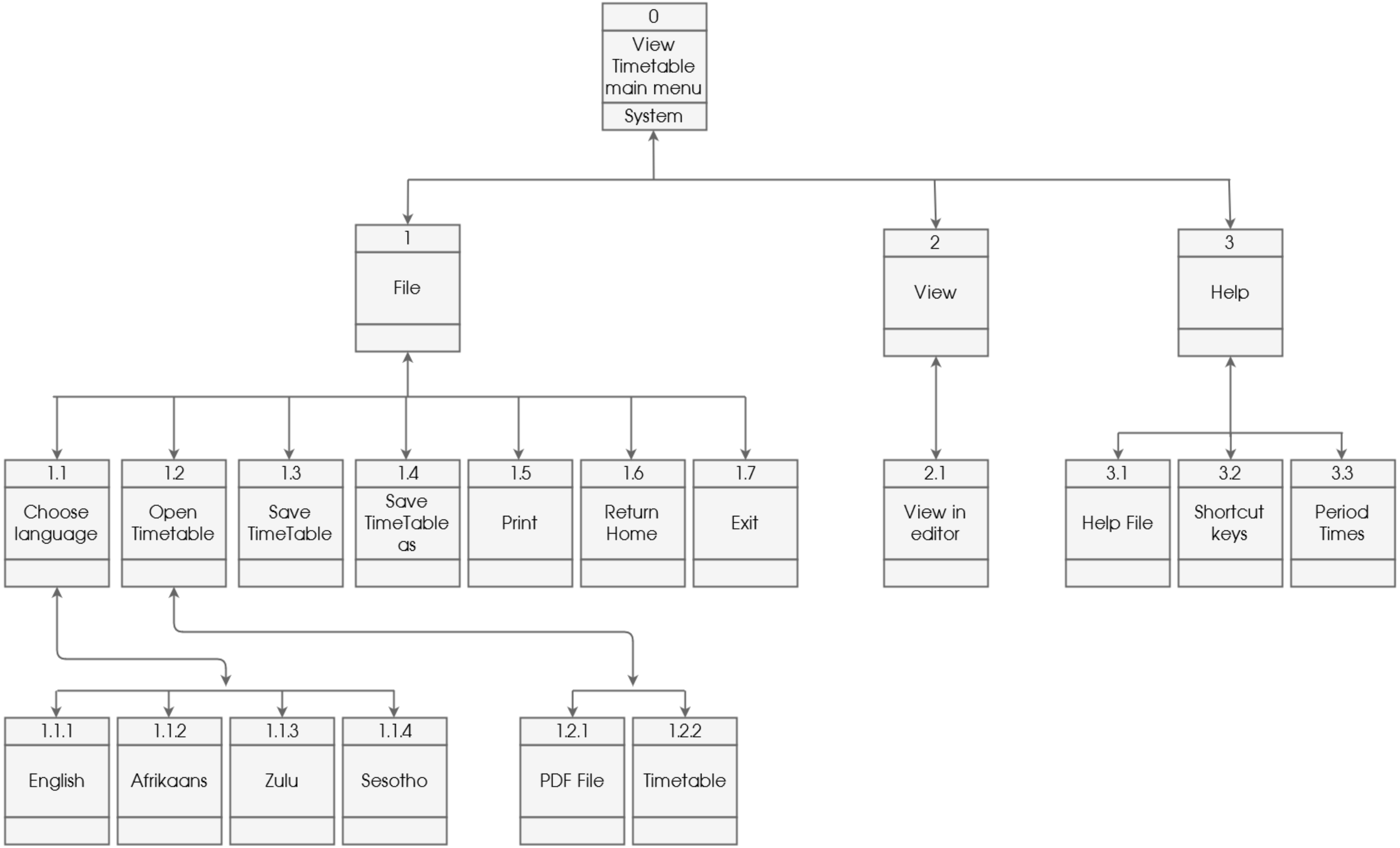
ADMIN HOME SCREEN MENU

An administrator using OmniCal would see the following menu on the home screen:



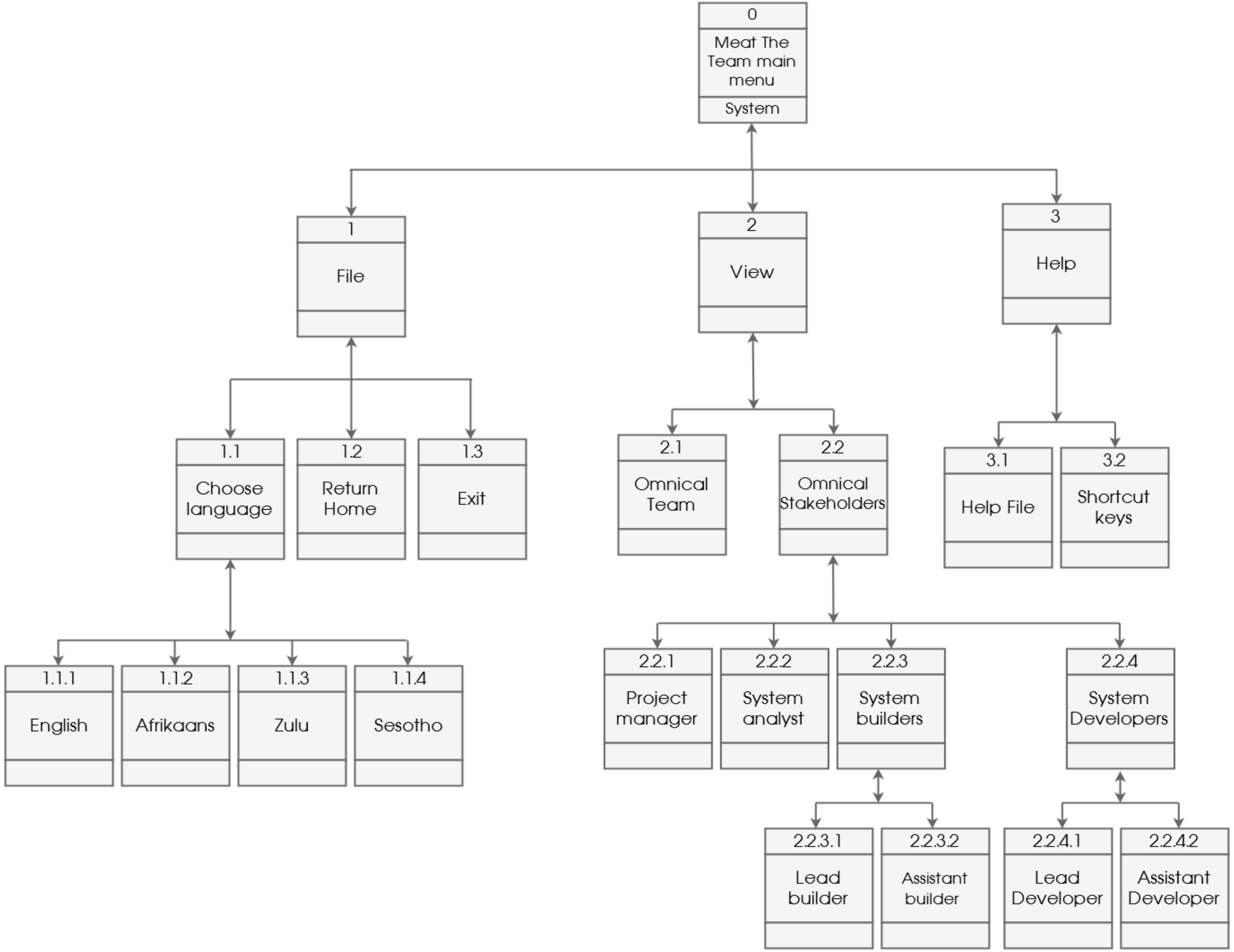
VIEW TIMETABLE MAIN MENU

OmniCal's view timetable form has the following main menu layout:



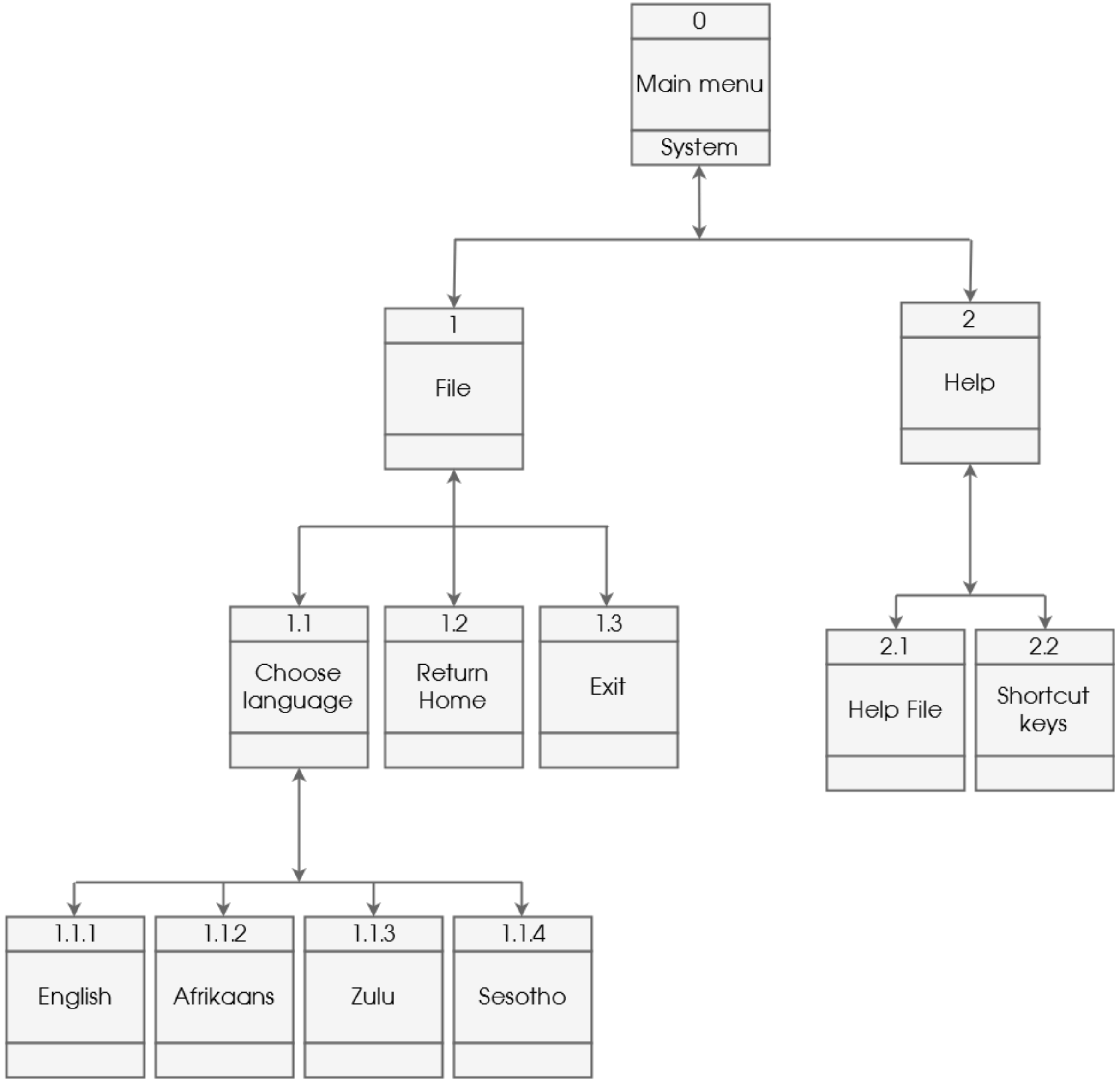
MEET THE TEAM MAIN MENU

OmniCal's meet the team form has the following main menu layout:



CUSTOM MAIN MENU

OmniCal has 5 forms as listed in the dialog chart list to which this custom main menu applies:



OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2:	ITRW212 12 G01	1-2:	ACCS111-1 12 G01	1-2:	ITRW214 3-103	1-2:	ACCS111-1 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3-103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3 103	7-8:	ITRW212 3 103

REFINING U/C MODEL

OmniCaL

BSCIT SECOND YEAR FIRST SEMESTER

Monday:	Tuesday:	Wednesday:	Thursday:
1-2: ITRW212 12 G01	1-2: ACCS111-1 25 G01	1-2: ITRW214 3 103	1-2: ACCS111-1 12 G05
5-6: ITRW211 3 103	5-6: ITRW212 3 103	3-4: ITRW214 3 103	3-4: ITRW211 9B G02
9-10: WVNS211 25 G08	7-8: ITRW212 3 103	5-6: ITRW213 3 103	5-6: ITRW212 3 103
11-12: ITRW213 25 G01		7-8: ITRW213 3 103	7-8: ITRW212 3 103

USE CASE LIST

Below is a list of Use Cases that our system makes use of:

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
Log in	Data is now accessible and editable
Log out	Data is no longer accessible and editable
Sign up	Information is created in order for log in to occur
Create event	Data is created to be sent to the database.
View timetable	Data is accessed and can be viewed
If admin	All data is accessible and editable
If student	Data for only that student can be accessed and edited
If worker	Data for only that worker can be accessed and edited
Help file	No data used
If English	Data is translated to English
If Afrikaans	Data is translated to Afrikaans
If Zulu	Data is translated to Zulu
If Sesotho	Data is translated to Sesotho
Close program	All data is released

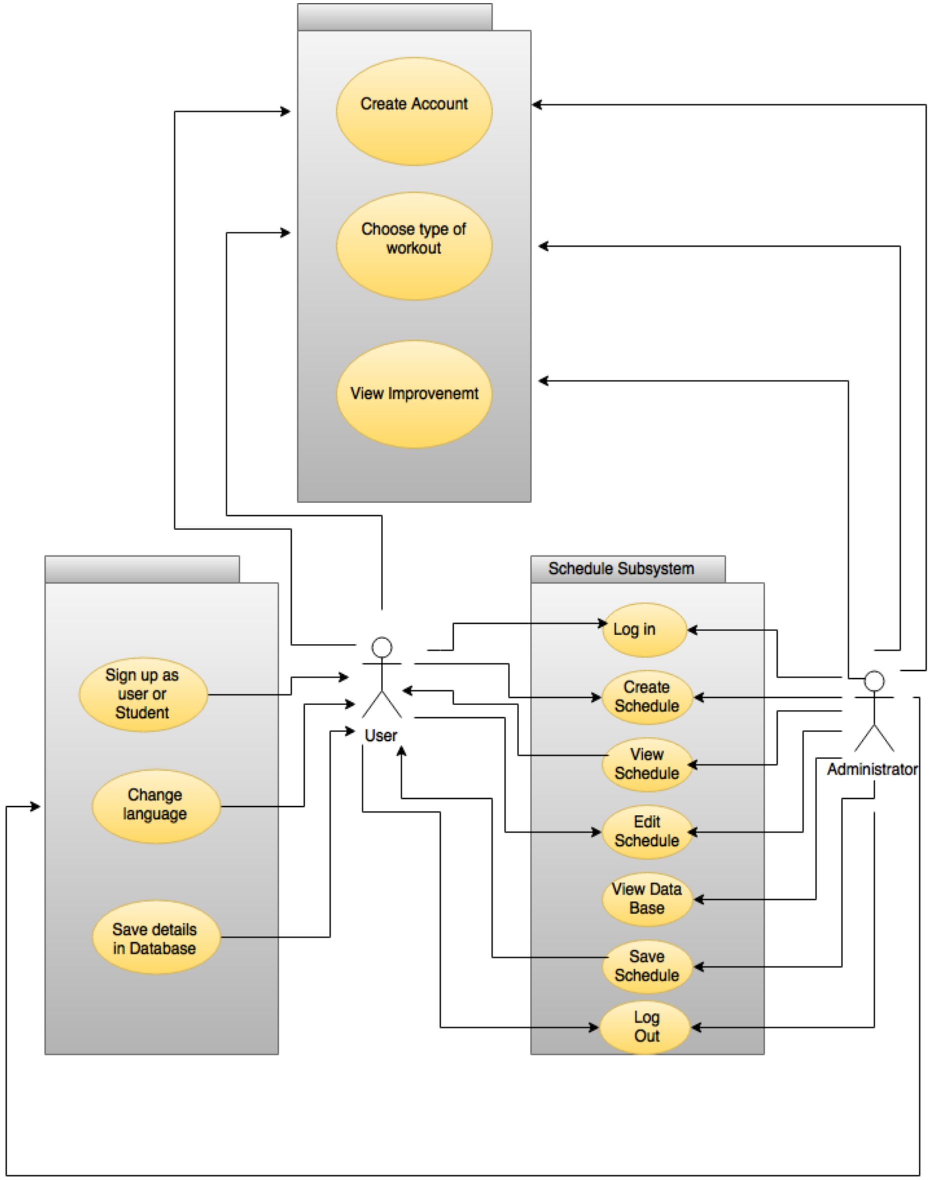
STEP 1: TRANSFROM FROM ANALYSIS TO DESIGN

Below is a list of use cases that have been refined for our system:

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

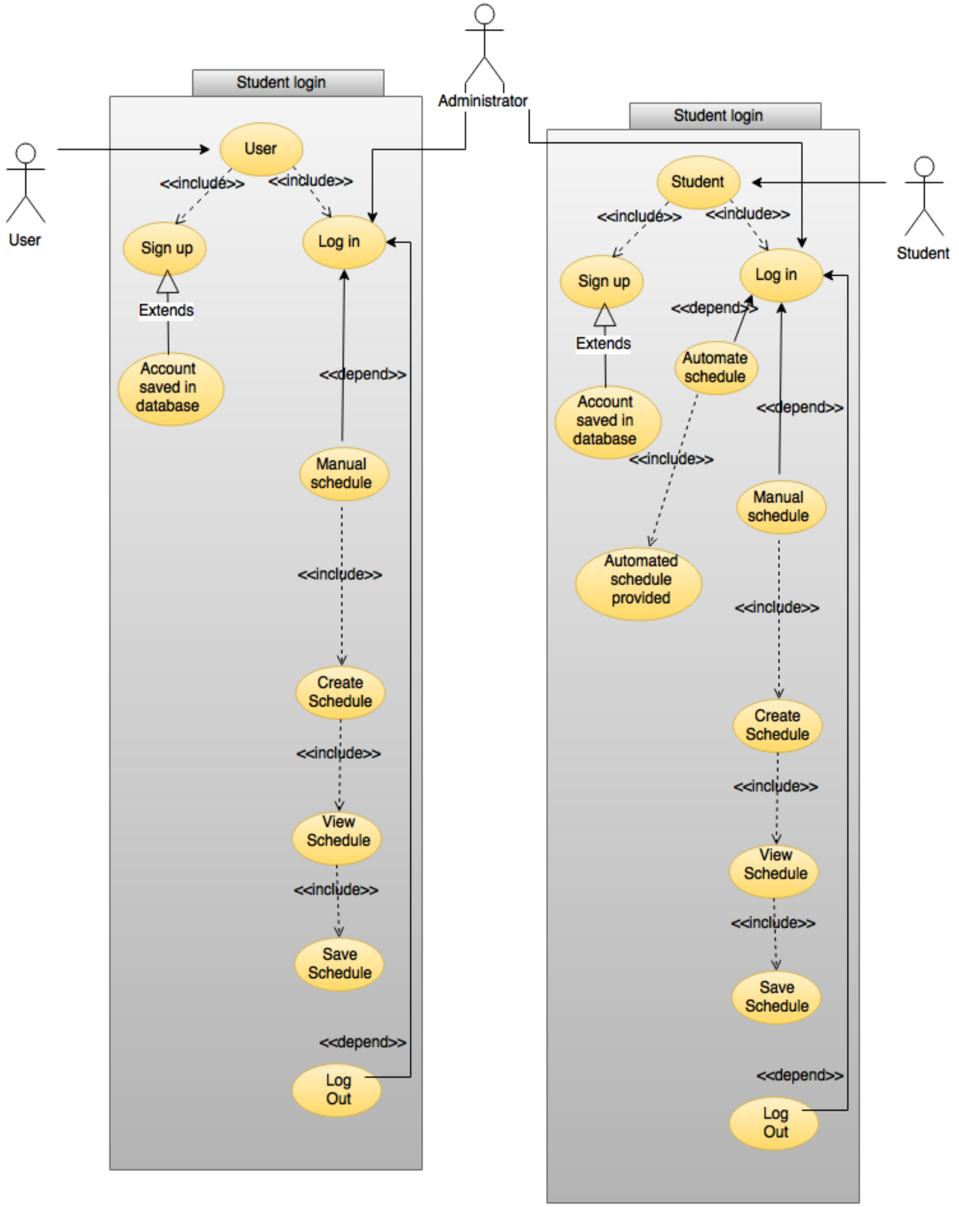
STEP 2: UPDATE USE CASE DIAGRAMS

Below is the use case diagram which reflects the interaction of our system:



STEP 2: UPDATE USE CASE DIAGRAMS (Continued)

Below is the use case diagram which reflects the depends on, extends and includes interaction of our system:



MODELING CLASS INTERACTIONS, BEHAVIORS AND SUPPORTING STATES

OmniCaL

BSCIT SECOND YEAR FIRST SEMESTER

MONDAY:	TUESDAY:	WEDNESDAY:	THURSDAY:
1-2: ITRW212 12 G01	1-2: ACCS111 12 G01	1-2: ITRW214 3 103	[illegible] ACCS111 12 G05
5-6: ITRW211 3 103	5-6: ITRW212 3 103	3-4: ITRW214 3 103	3-4: ITRW211 9B G02
9-10: WVNS211 25 G08	7-8: ITRW212 3 103	5-6: ITRW213 3 103	5-6: ITRW212 3 103
11-12: ITRW213 25 G01		7-8: ITRW213 3 103	7-8: ITRW212 3 103

STEP 1: IDENTIFY AND CLARIFY USE CASE DESIGN CLASSES

The redefined use cases were used and expanded to clarify the design classes

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

STEP 2: IDENTIFY CLASS ATTRIBUTES

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.

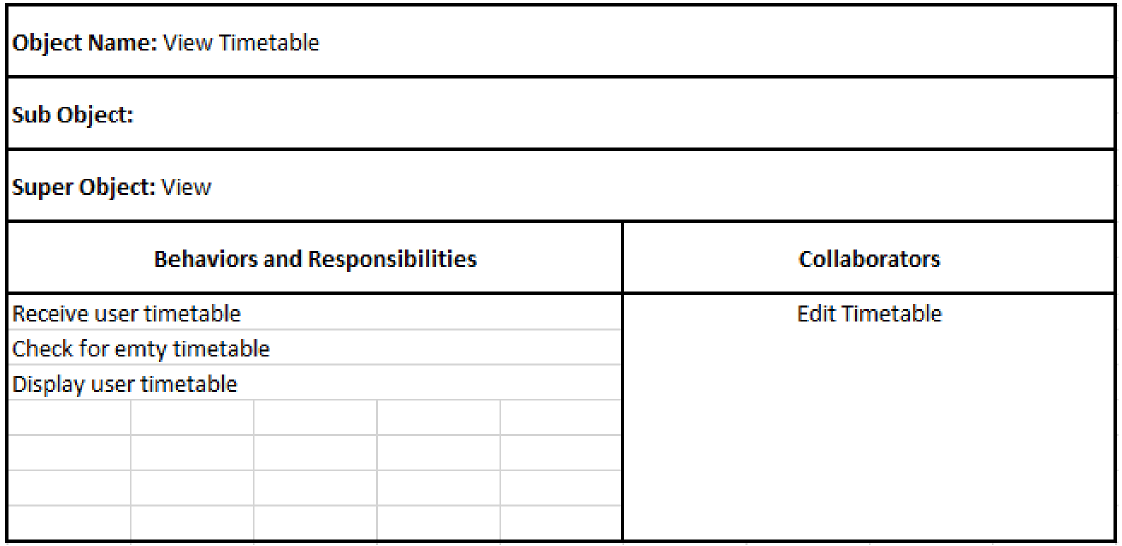
STEP 3: IDENTIFY CLASS BEHAVIOURS AND RESPONSIBILITIES

Below is a partial summary of use case behaviours

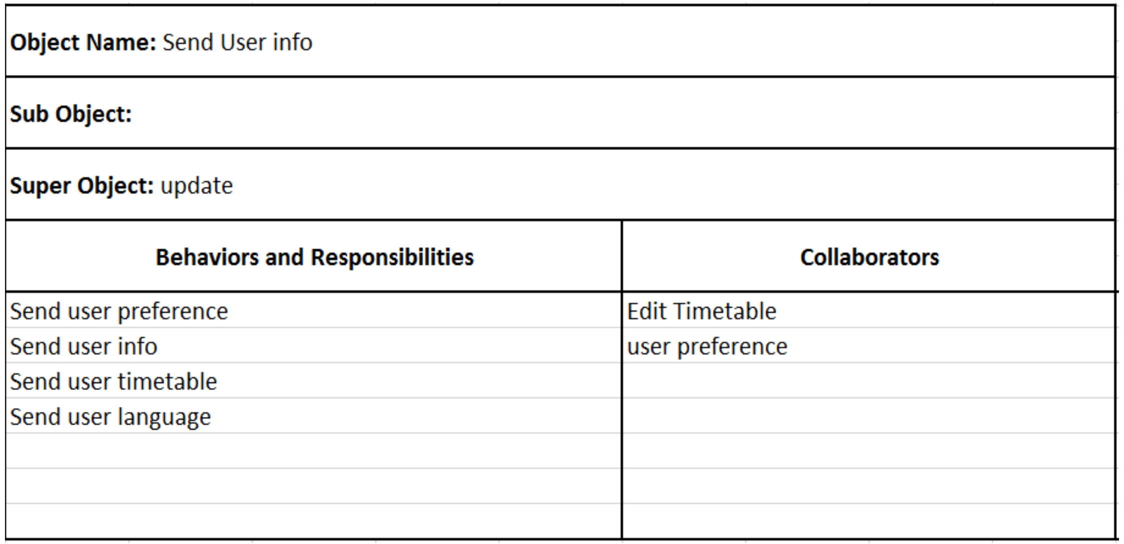
Behaviors	Automated/manual	Class type
Receive timetable	Automated	Entity
Check if empty	Automated	Entity
Display Timetable	Automated	Entity
Send user preference	Manual	Entity
Send user info	Manual	Controller
Send user timetable	Manual	Entity
Send user language	Manual	Entity
Retrieve user preferences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

STEP 3: IDENTIFY CLASS BEHAVIOURS AND RESPONSIBILITIES

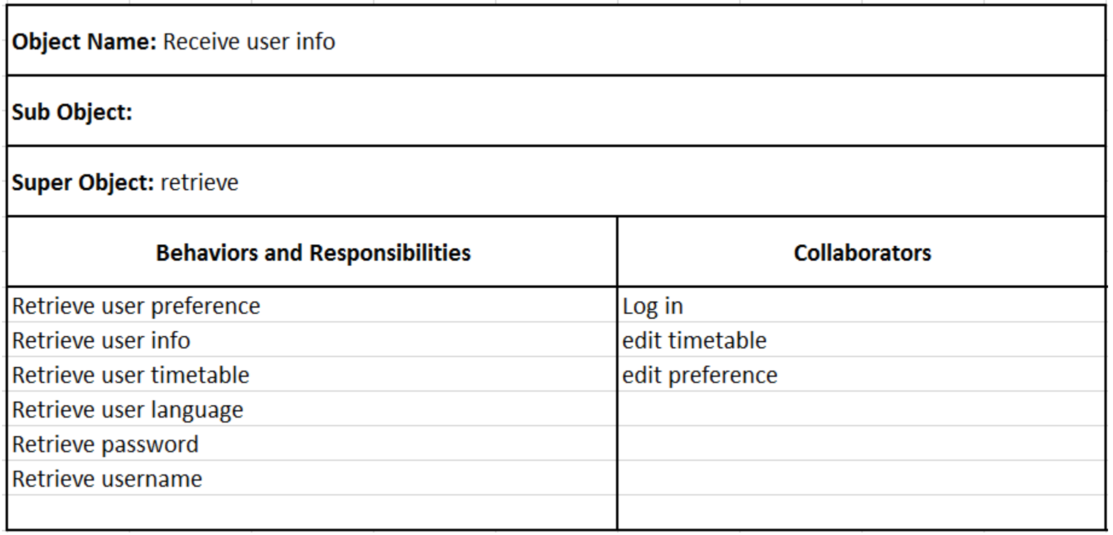
Below is a partial summary of use case behaviours and responsibilities in the form of CRC cards



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	



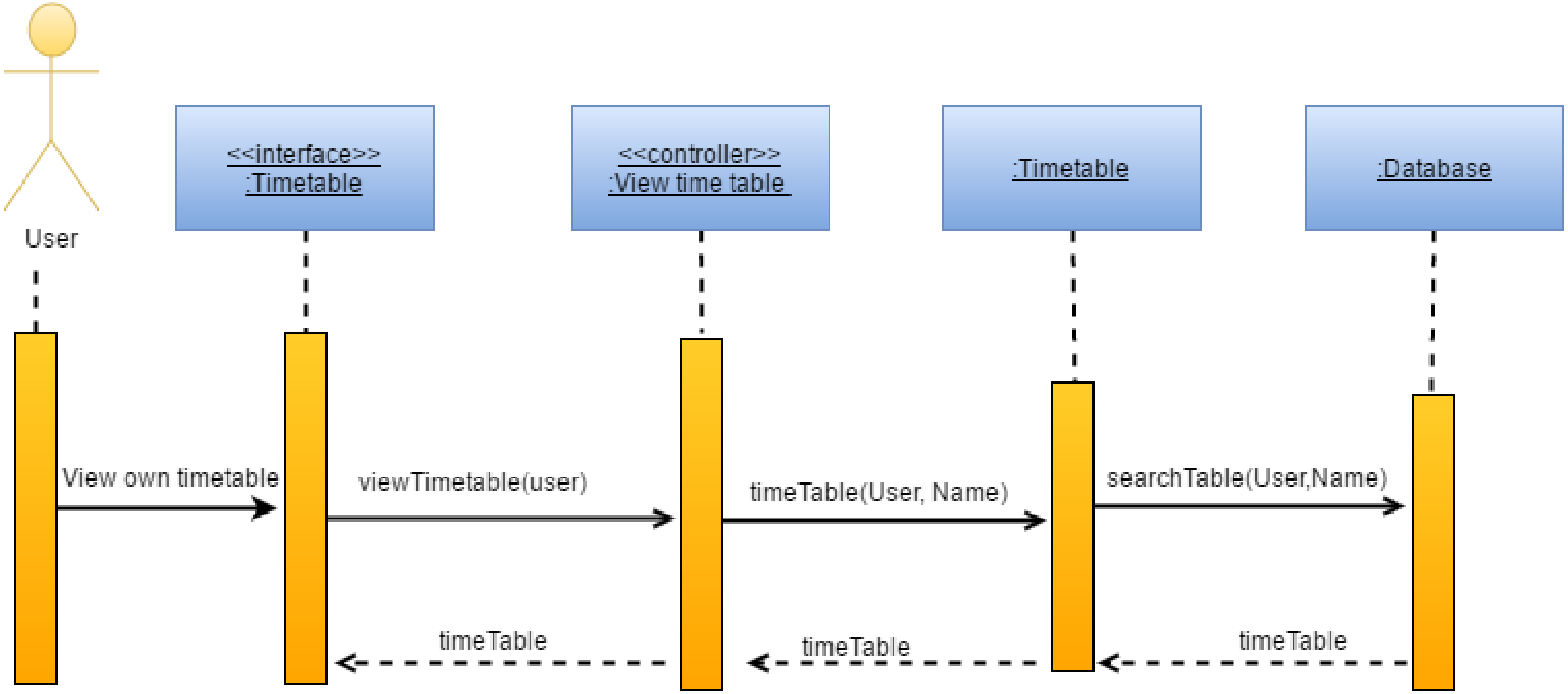
Object Name: Send User info	
Sub Object:	
Super Object: update	
Behaviors and Responsibilities	**Collaborators**
Send user preference	Edit Timetable
Send user info	user preference
Send user timetable	
Send user language	



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	

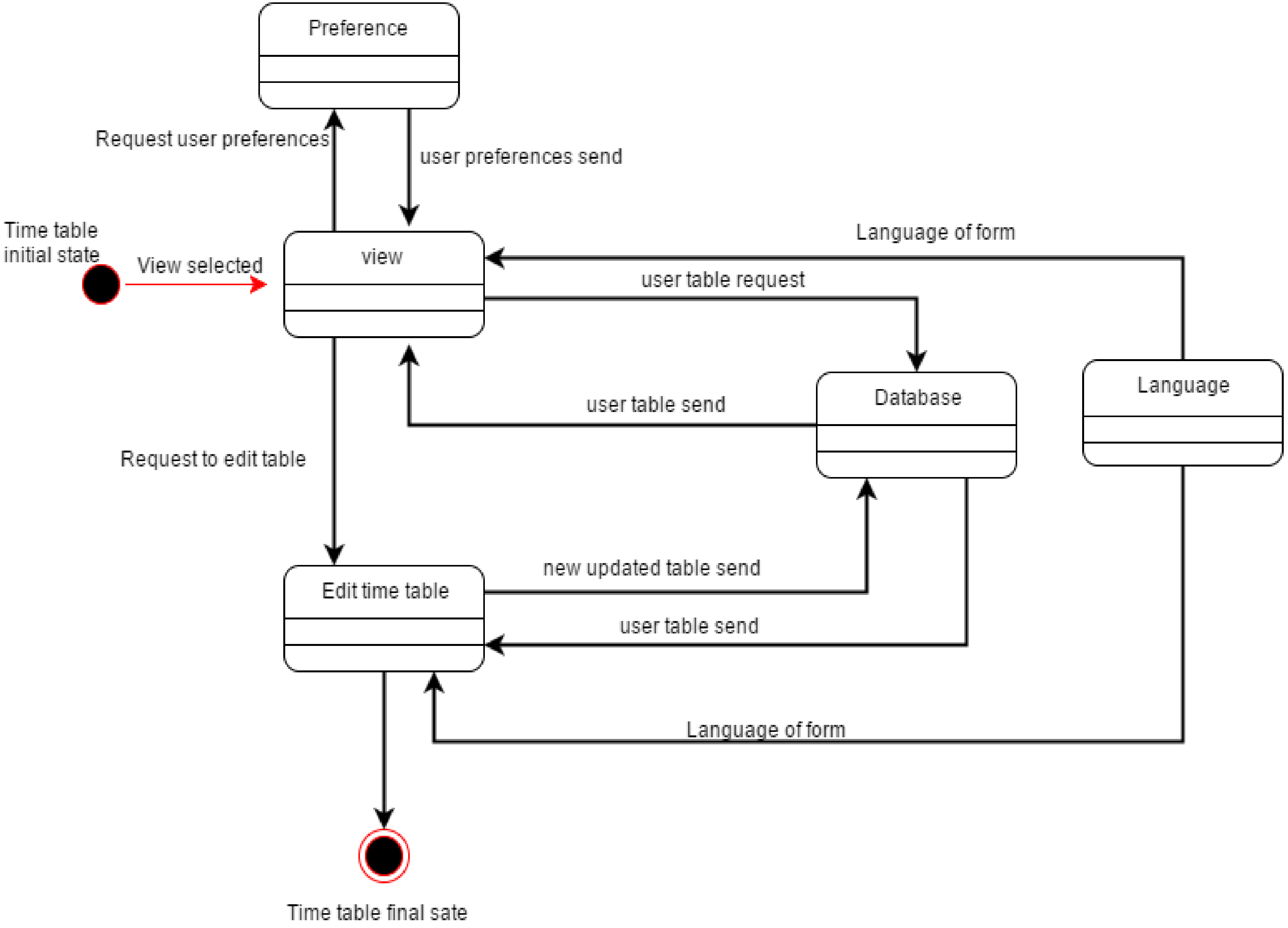
STEP 3: IDENTIFY CLASS BEHAVIOURS AND RESPONSIBILITIES

Below is the sequence diagram of our system:



STEP 4: MODEL OBJECT STATES

Below is the Statechart diagram of our system:



UPDATING THE OBJECT MODEL

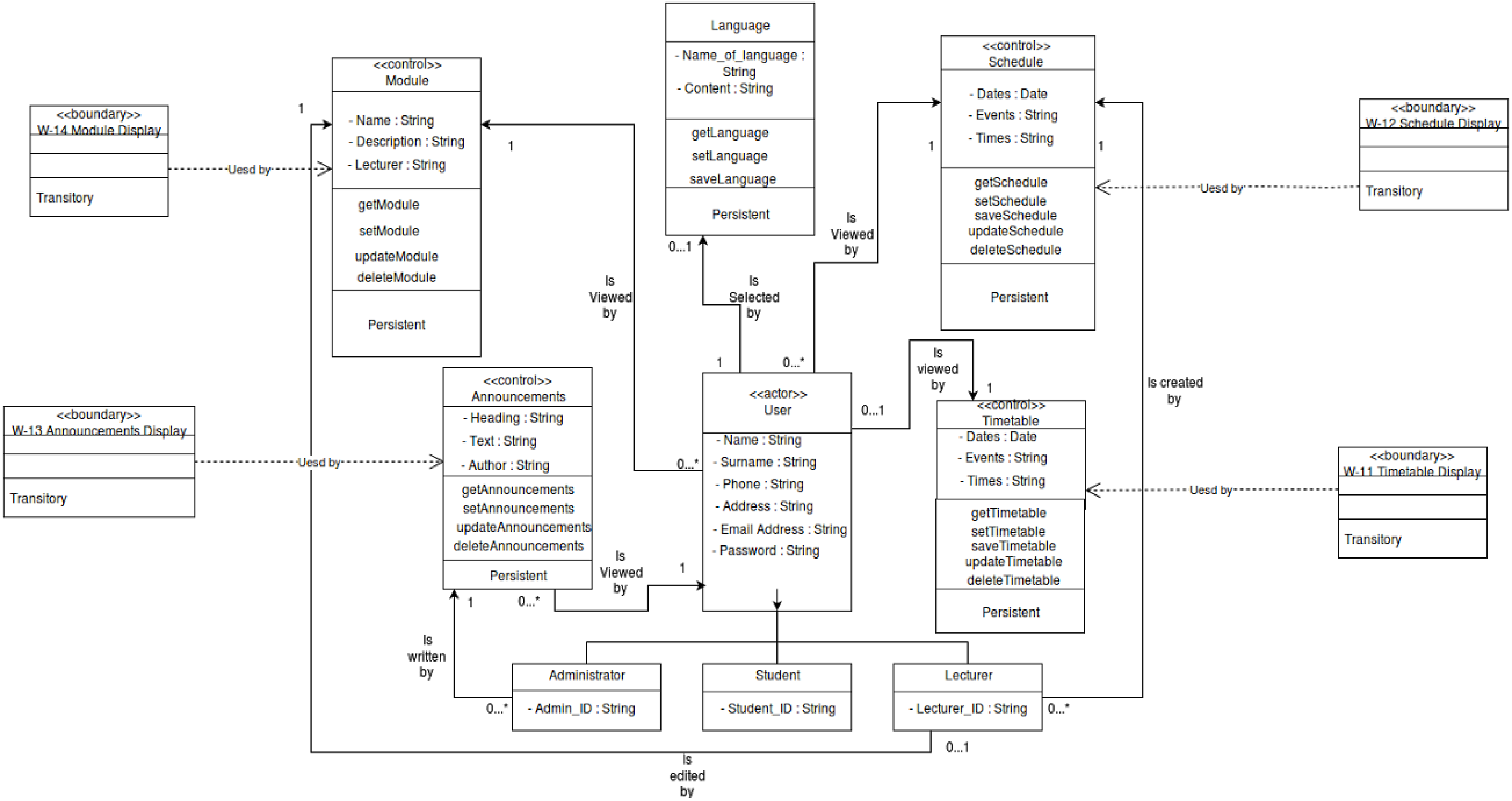
OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2:	ITRW212 12 G01	1-2:	ACCS111-1 25 G01	1-2:	ITRW214 3 103	1-2:	ACCS111-1 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3 103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3 103	7-8:	ITRW212 3 103

STEP 4: MODEL OBJECT STATES

Below is the partial design class diagram of our systems use cases:



<<boundary>>
W-14 Module Display
Transitory

<<control>>
Module
- Name : String
- Description : String
- Lecturer : String
getModule
setModule
updateModule
deleteModule
Persistent

Language
- Name_of_language : String
- Content : String
getLanguage
setLanguage
saveLanguage
Persistent

<<control>>
Schedule
- Dates : Date
- Events : String
- Times : String
getSchedule
setSchedule
saveSchedule
updateSchedule
deleteSchedule
Persistent

<<boundary>>
W-12 Schedule Display
Transitory

<<boundary>>
W-13 Announcements Display
Transitory

<<control>>
Announcements
- Heading : String
- Text : String
- Author : String
getAnnouncements
setAnnouncements
updateAnnouncements
deleteAnnouncements
Persistent

<<actor>>
User
- Name : String
- Surname : String
- Phone : String
- Address : String
- Email Address : String
- Password : String

<<control>>
Timetable
- Dates : Date
- Events : String
- Times : String
getTimetable
setTimetable
saveTimetable
updateTimetable
deleteTimetable
Persistent

<<boundary>>
W-11 Timetable Display
Transitory

Administrator
- Admin_ID : String

Student
- Student_ID : String

Lecturer
- Lecturer_ID : String

DESIGN PATTERNS

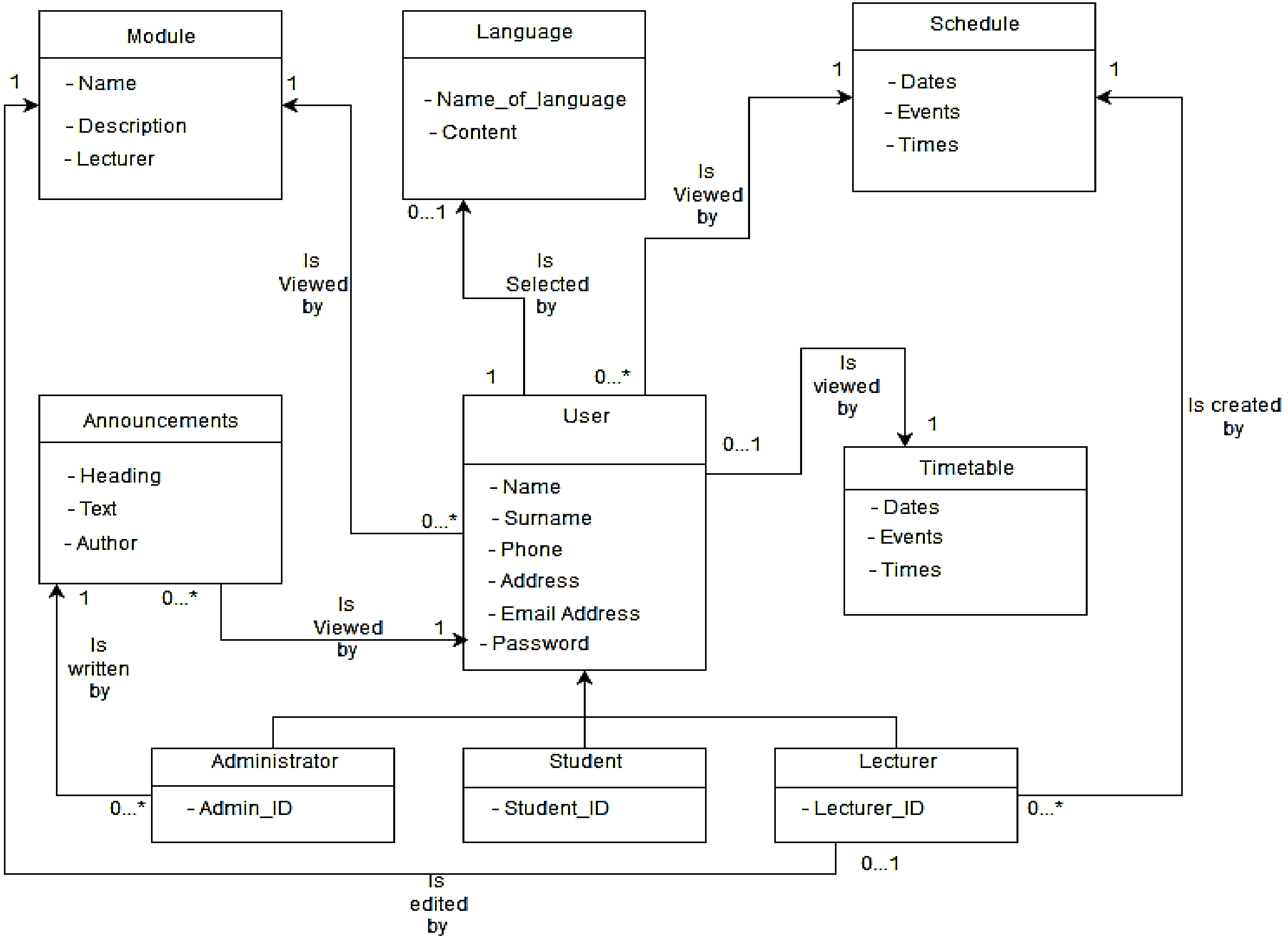
OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:	Tuesday:	Wednesday:	Thursday:
1-2: ITRW212 12 G01	1-2: ACCS111 25 G01	1-2: ITRW214 3 103	1-2: ACCS111 12 G05
5-6: ITRW211 3 103	5-6: ITRW212 3 103	3-4: ITRW214 3 103	3-4: ITRW211 9B G02
9-10: WVNS211 25 G08	7-8: ITRW212 3 103	5-6: ITRW213 3 103	5-6: ITRW212 3 103
11-12: ITRW213 25 G01		7-8: ITRW213 3 103	7-8: ITRW212 3 103

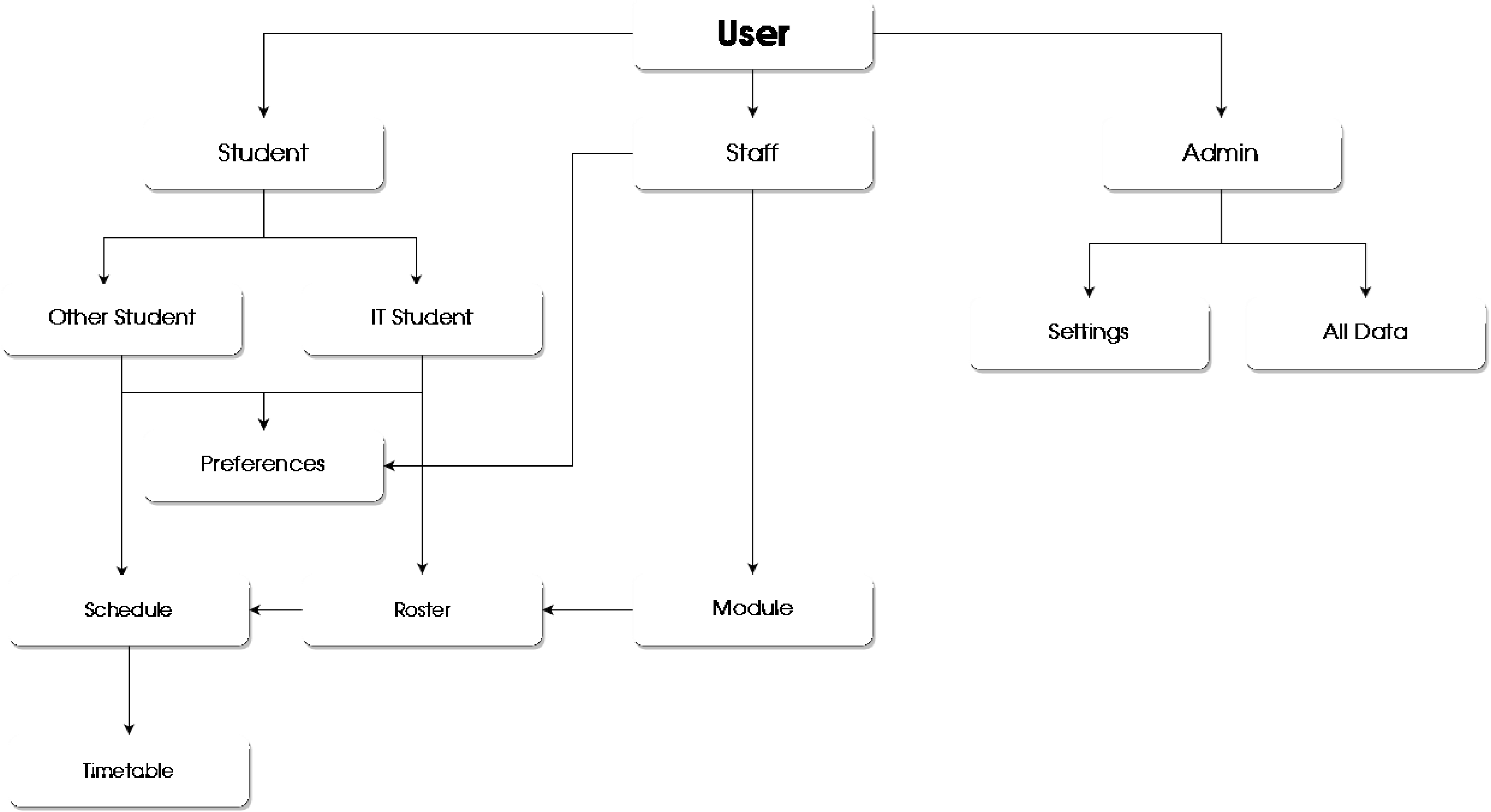
ADAPTER PATTERN

Below is the adapter pattern of our system:



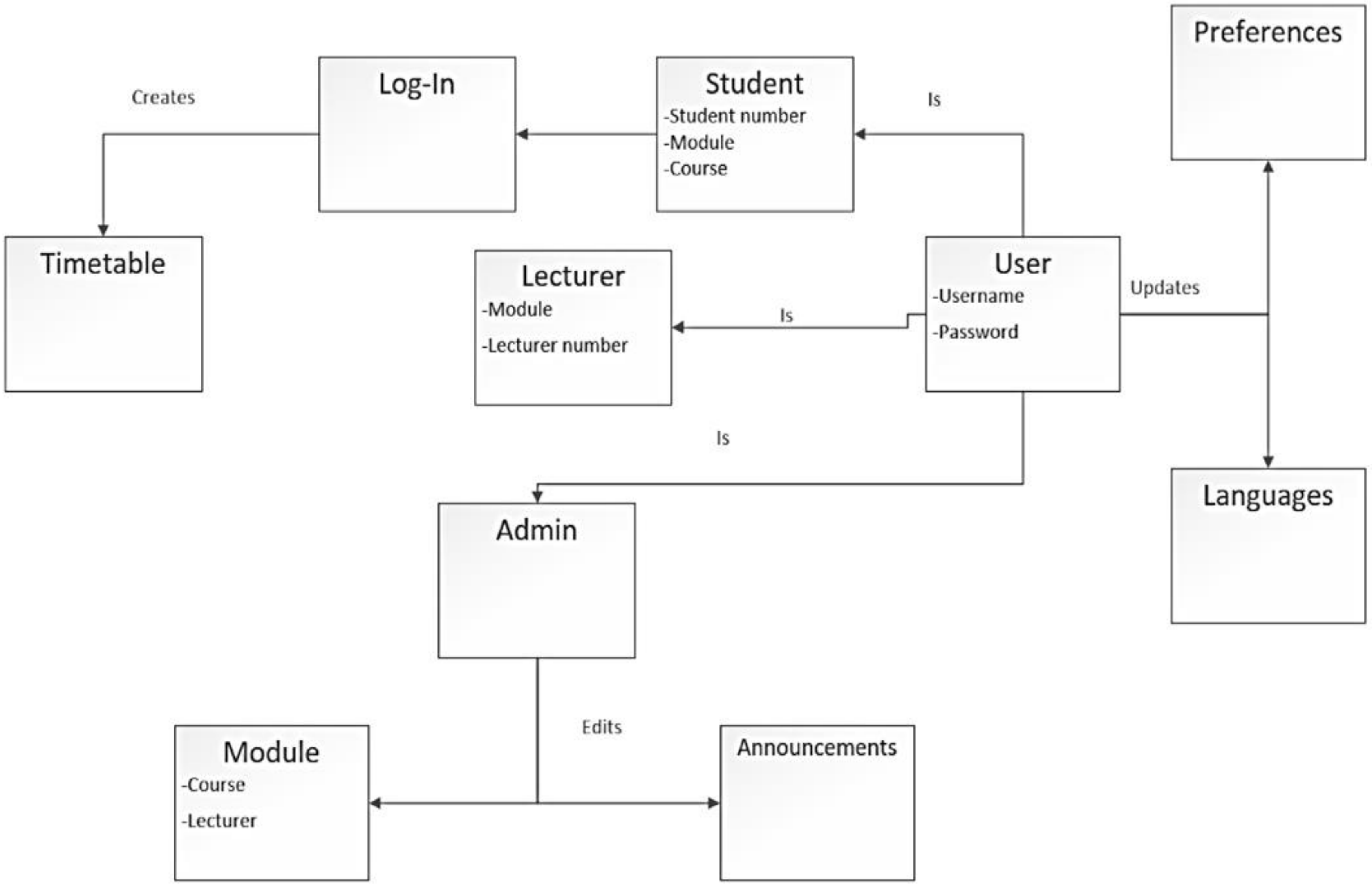
ORGANISATIONAL PATTERN

Below is the organisational structure of our system:



STRATEGY PATTERN

Below is the strategy pattern of our system:

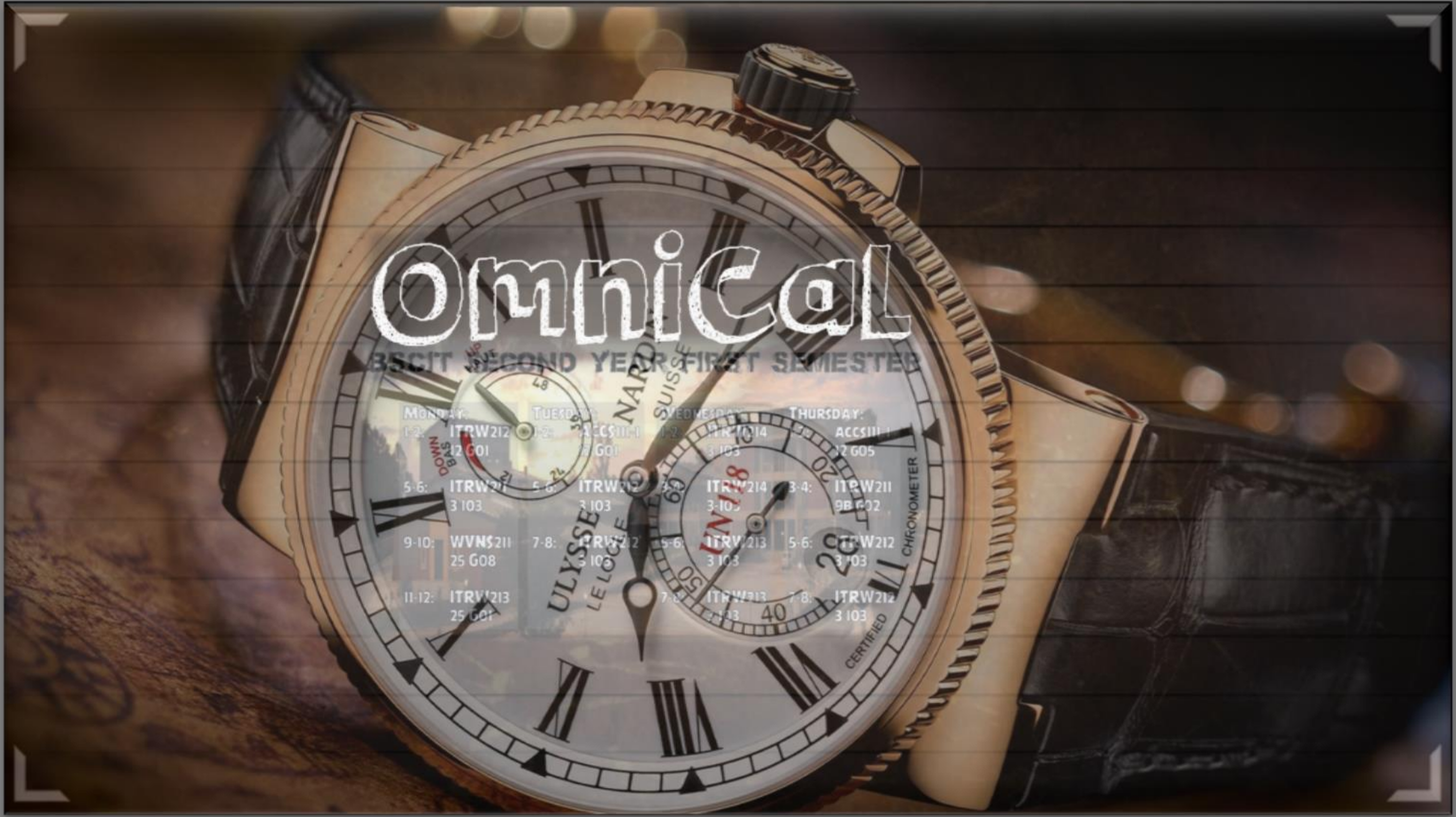


OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2.	ITRW212 12 G01	1-2:	ACCS111-1 12 G01	1-2:	ITRW214 3-103	1-2:	ACCS111-1 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3-103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3-103	7-8:	ITRW212 3 103

IMPLEMENTATION PHASE



OmniCaL

BSCIT SECOND YEAR FIRST SEMESTER

Monday:		Tuesday:		Wednesday:		Thursday:	
1-2:	ITRW212 12 G01	1-2:	ACCS111-1 12 G01	1-2:	ITRW214 3 103	[illegible]	ACCS111-1 12 G05
5-6:	ITRW211 3 103	5-6:	ITRW212 3 103	3-4:	ITRW214 3 103	3-4:	ITRW211 9B G02
9-10:	WVNS211 25 G08	7-8:	ITRW212 3 103	5-6:	ITRW213 3 103	5-6:	ITRW212 3 103
11-12:	ITRW213 25 G01			7-8:	ITRW213 3 103	7-8:	ITRW212 3 103

CONDUCT SYSTEM TEST

OmniCal's system test results are laid out below:

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

CONDUCT SYSTEM TEST (Continued)

Test Data:

- Data stored in textfiles for various language options
  - omnical licence agreement Afrikaans
  - omnical licence agreement English
  - omnical licence agreement 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
  - frmUpdateProfile Afrikaans
  - frmUpdateProfile English
  - frmUpdateProfile Zulu

- Data stored in database in various tables:
  - administration
  - course
  - module
  - period
  - staff
  - student

CONDUCT SYSTEM TEST (Concluded)

Problems and issues

- Without internet connection, the database cannot be accessed

Problems revealed during testing

- Internet connection may be a problem

Verification of system operation

- The parts of the system that operate with test data operate correctly

PREPARE CONVERSION PLAN

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.

INSTALL DATABASES

The following databases have been installed:

OmniCal Database installed	☑

TRAIN USERS

User help/instruction file

Once the program is opened you (the user) have a choice of the following:
1. Enter user type and details to login (home)
2. Choose to register (Proceed to register form)
3. The user may at any point in the login screen click on file and change the language of the program

Here the user may enter their details and sign up to continue with the creation of the timetable.

This is the main hub of the program, linking all essential forms in one simple to use form. On the main screen of this form the user will see their timetable. On this screen the user may create, edit or remove any timetable.

CONVERT TO NEW SYSTEM

Our system conversion was successful and was converted from Delphi to C#

- We started hosting our MySQL Database online
- We created a mobile application
- The physical system and application is now available for download from our website:

  jacquijm92.wixsite.com/omnical/download

CONSTRUCTION PHASE

OmniCaL

BSCIT SECOND YEAR FIRST SEMESTER

Monday:
1-2: ITRW212 12 G01
5-6: ITRW211 3 103
9-10: WVNS211 25 G08
11-12: ITRW213 25 G01

Tuesday:
1-2: ACCS111 25 G01
5-6: ITRW212 3 103
7-8: ITRW212 3 103

Wednesday:
1-2: ITRW214 3 103
3-4: ITRW214 3 103
5-6: ITRW213 3 103
7-8: ITRW213 3 103

Thursday:
1-2: ACCS111 12 G05
3-4: ITRW211 9B G02
5-6: ITRW212 3 103
7-8: ITRW212 3 103

BUILD AND TEST NETWORKS

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:

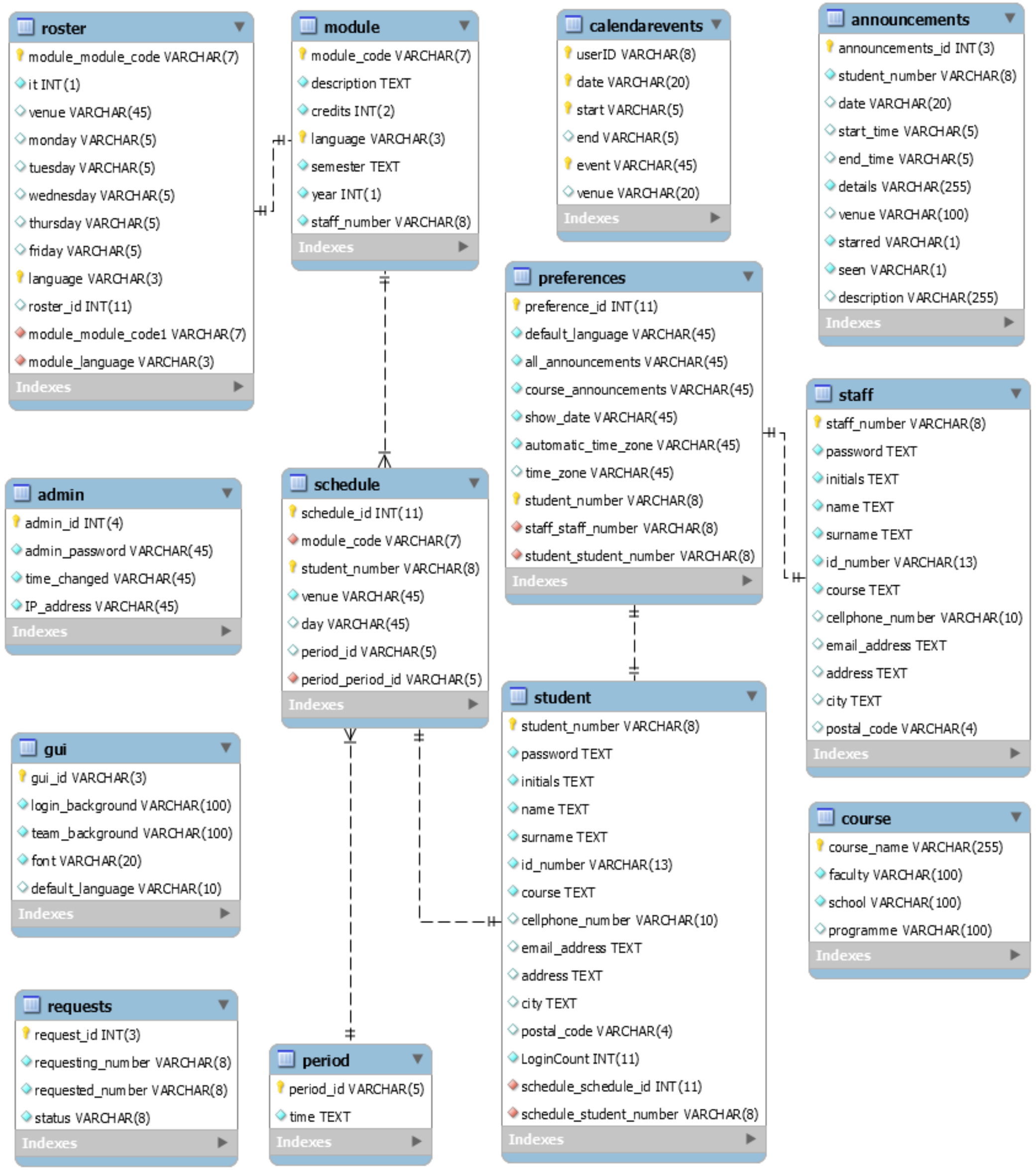
- Network Adapters
- USB Adapters for User input
- Mouse Devices
- Printers
- Monitor
- Touchpad

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

BUILD AND TEST DATABASES

OmniCal makes use of one database, created in MySQL



roster
module_module_code VARCHAR(7)
it INT(1)
venue VARCHAR(45)
monday VARCHAR(5)
tuesday VARCHAR(5)
wednesday VARCHAR(5)
thursday VARCHAR(5)
friday VARCHAR(5)
language VARCHAR(3)
roster_id INT(11)
module_module_code1 VARCHAR(7)
module_language VARCHAR(3)
Indexes

module
module_code VARCHAR(7)
description TEXT
credits INT(2)
language VARCHAR(3)
semester TEXT
year INT(1)
staff_number VARCHAR(8)
Indexes

calendarevents
userID VARCHAR(8)
date VARCHAR(20)
start VARCHAR(5)
end VARCHAR(5)
event VARCHAR(45)
venue VARCHAR(20)
Indexes

announcements
announcements_id INT(3)
student_number VARCHAR(8)
date VARCHAR(20)
start_time VARCHAR(5)
end_time VARCHAR(5)
details VARCHAR(255)
venue VARCHAR(100)
starred VARCHAR(1)
seen VARCHAR(1)
description VARCHAR(255)
Indexes

preferences
preference_id INT(11)
default_language VARCHAR(45)
all_announcements VARCHAR(45)
course_announcements VARCHAR(45)
show_date VARCHAR(45)
automatic_time_zone VARCHAR(45)
time_zone VARCHAR(45)
student_number VARCHAR(8)
staff_staff_number VARCHAR(8)
student_student_number VARCHAR(8)
Indexes

staff
staff_number VARCHAR(8)
password TEXT
initials TEXT
name TEXT
surname TEXT
id_number VARCHAR(13)
course TEXT
cellphone_number VARCHAR(10)
email_address TEXT
address TEXT
city TEXT
postal_code VARCHAR(4)
Indexes

admin
admin_id INT(4)
admin_password VARCHAR(45)
time_changed VARCHAR(45)
IP_address VARCHAR(45)
Indexes

schedule
schedule_id INT(11)
module_code VARCHAR(7)
student_number VARCHAR(8)
venue VARCHAR(45)
day VARCHAR(45)
period_id VARCHAR(5)
period_period_id VARCHAR(5)
Indexes

student
student_number VARCHAR(8)
password TEXT
initials TEXT
name TEXT
surname TEXT
id_number VARCHAR(13)
course TEXT
cellphone_number VARCHAR(10)
email_address TEXT
address TEXT
city TEXT
postal_code VARCHAR(4)
LoginCount INT(11)
schedule_schedule_id INT(11)
schedule_student_number VARCHAR(8)
Indexes

gui
gui_id VARCHAR(3)
login_background VARCHAR(100)
team_background VARCHAR(100)
font VARCHAR(20)
default_language VARCHAR(10)
Indexes

course
course_name VARCHAR(255)
faculty VARCHAR(100)
school VARCHAR(100)
programme VARCHAR(100)
Indexes

requests
request_id INT(3)
requesting_number VARCHAR(8)
requested_number VARCHAR(8)
status VARCHAR(8)
Indexes

period
period_id VARCHAR(5)
time TEXT
Indexes

INSTALL AND TEST NEW SOFTWARE

The following software has been installed and tested by the systems analyst and the project manager

- New Software
    - MySQL – Installed and tested
    - PDF – Installed and tested

WRITE AND TEST NEW PROGRAM

Writing and testing of the program commenced and was executed through C#

OmniCal

BSCIT SECOND YEAR FIRST SEMESTER

Monday:	Tuesday:	Wednesday:	Thursday:
1-2. ITRW212 12 G01	1-2: ACCS111-1 12 G01	1-2: ITRW214 3-103	1-2: ACCS111-1 12 G05
5-6: ITRW211 3 103	5-6: ITRW212 3 103	3-4: ITRW214 3-103	3-4: ITRW211 9B G02
9-10: WVNS211 25 G08	7-8: ITRW212 3 103	5-6: ITRW213 3 103	5-6: ITRW212 3 103
11-12: ITRW213 25 G01		7-8: ITRW213 3 103	7-8: ITRW212 3 103

OPERATION AND SUPPORT CHECKLIST

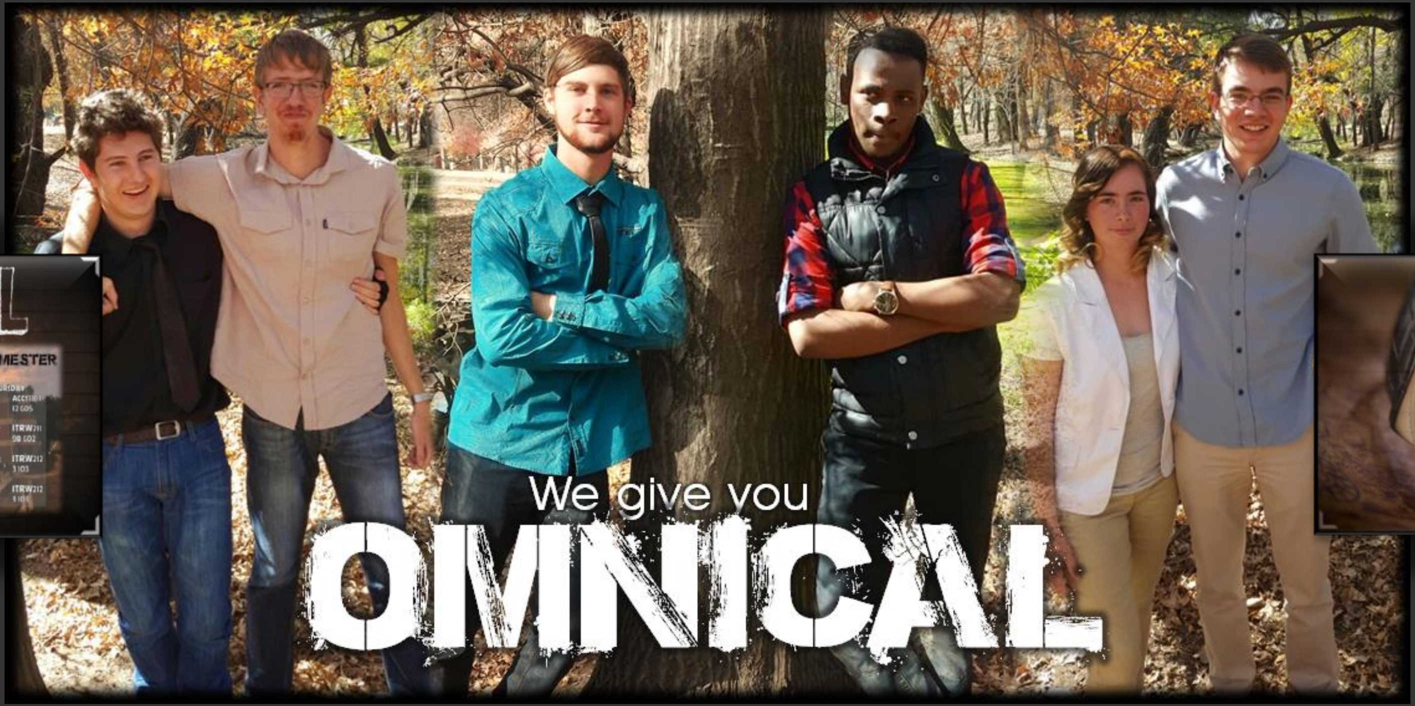
SYSTEM OPERATIONS AND SUPPORT CHECKLIST

The following represents the checklist we used to complete our system operation and support

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

THANK YOU!

OmniCal
BSCIT SECOND YEAR FIRST SEMESTER
Calendar 2016
MONDAY
TUESDAY
WEDNESDAY
THURSDAY


We give you
OMNICAL

OmniCal