CIS020-2 – Systems Development and Modern Database Practices

Assignment 2 – Student Information Kiosk (SIK) Case Study - Group Assignment

Group Number – Group 12

1814927 – Michael Songer

Contents

[Team member roles 2](#_Toc41159139)

[Introduction 2](#_Toc41159140)

[Plan of work 2](#_Toc41159141)

[Role identification 2](#_Toc41159142)

[Users 2](#_Toc41159143)

[Client/stakeholder 2](#_Toc41159144)

[Service/audience 3](#_Toc41159145)

[Soft Systems Methodology (SSM) 3](#_Toc41159146)

[Rich Picture 3](#_Toc41159147)

[Root Definition 3](#_Toc41159148)

[CATWOE 3](#_Toc41159149)

[Conceptual Data Model 3](#_Toc41159150)

[The three E’s 3](#_Toc41159151)

[Requirements 3](#_Toc41159152)

[Use case diagram 5](#_Toc41159153)

[Activity diagram 6](#_Toc41159154)

[Class diagram 8](#_Toc41159155)

[Entity relationship diagram 8](#_Toc41159156)

[Skeleton Tables 8](#_Toc41159157)

[Entity Relationships 8](#_Toc41159158)

[Database Design 8](#_Toc41159159)

[Tables 8](#_Toc41159160)

[SQL 9](#_Toc41159161)

[User-interface design 10](#_Toc41159162)

[Security 11](#_Toc41159163)

[Implementation of kiosk 11](#_Toc41159164)

[Innovations 11](#_Toc41159165)

[Test system 11](#_Toc41159166)

[Future evolutions 11](#_Toc41159167)

[Conclusion 11](#_Toc41159168)

[References 11](#_Toc41159169)

[Appendix 11](#_Toc41159170)

# Team member roles

1814927 - Michael songer

Team manager, group report, introduction, plan of work, role identification, soft system methodology, requirements, database design, user-interface design, security.

Zaid

Use case diagram, activity diagram,

# Introduction

The goal of the assignment is to design and make a student information kiosk that will be a useful tool for university students. It should have a variety of tools that informs students about different what’s happening both in the university and locally.

# Plan of work

The group originally started as six people but ended up going down to two, making doing the work very difficult. So, we tried to split the work evenly. Michael was set as group leader, but assigning the work was agreed together. Michael was to come up with the requirements and design the soft system methodologies. From the requirements Zaid was to do the UML diagrams. Michael then was to work on the database design and GUI design. Once all the designing was done and agreed upon it will be implemented into oracle apex.

# Role identification

## Users

Students, university staff and admin. The university staff and admin will be the ones who add things like event details and local points of interest.

## Client/stakeholder

University dean, university students, university staff, university event management team, university administration.

## Service/audience

The service is an online kiosk that is designed to provide information to students of the university of Bedfordshire. The information will be things like local places of interest, events (both on campus and near). It will also inform the students of where they can get support if they require it.

# Soft Systems Methodology (SSM)

## Rich Picture

## Root Definition

The student information kiosk is made to easily show students anything that will be of interest to them (x), by means of a simple/easy to use online system(y), in order to aid them in getting a memorable university experience and meet other students with similar interest.

## CATWOE

C – Students

A – University staff

T – Increase popularity/knowledge of things students may be interested in.

W – Having students socialise together more will help their social skills for the future and having students at local events will increase people’s views on the university.

O – University dean and university event management team.

E – University budget, internet/mobile data connectivity, student devices, interest.

## Conceptual Data Model

## The three E’s

Efficacy -

Efficiency -

Effectiveness -

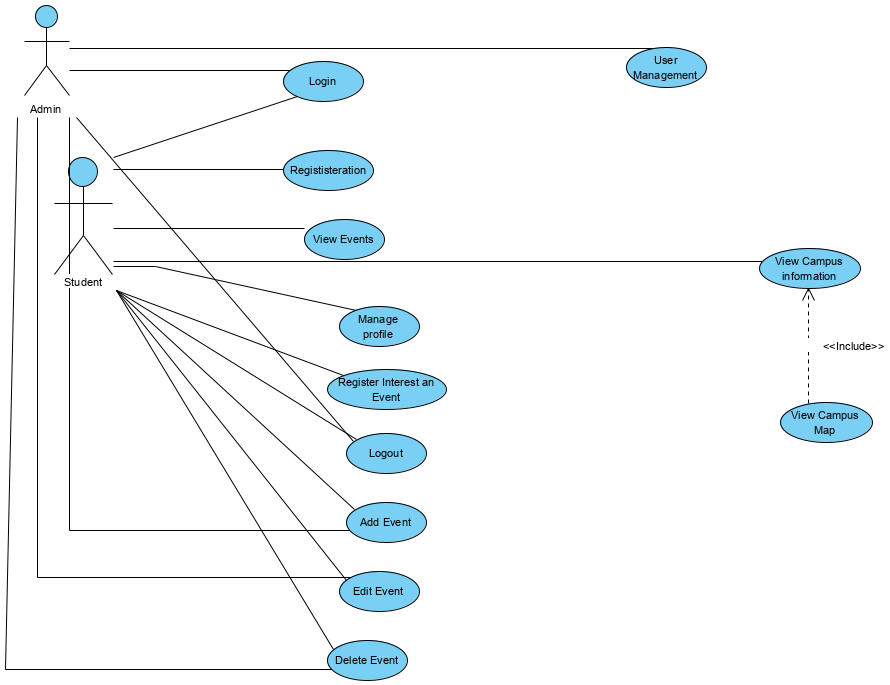
# Requirements

We were able to get an initial list of requirements from the clients brief.

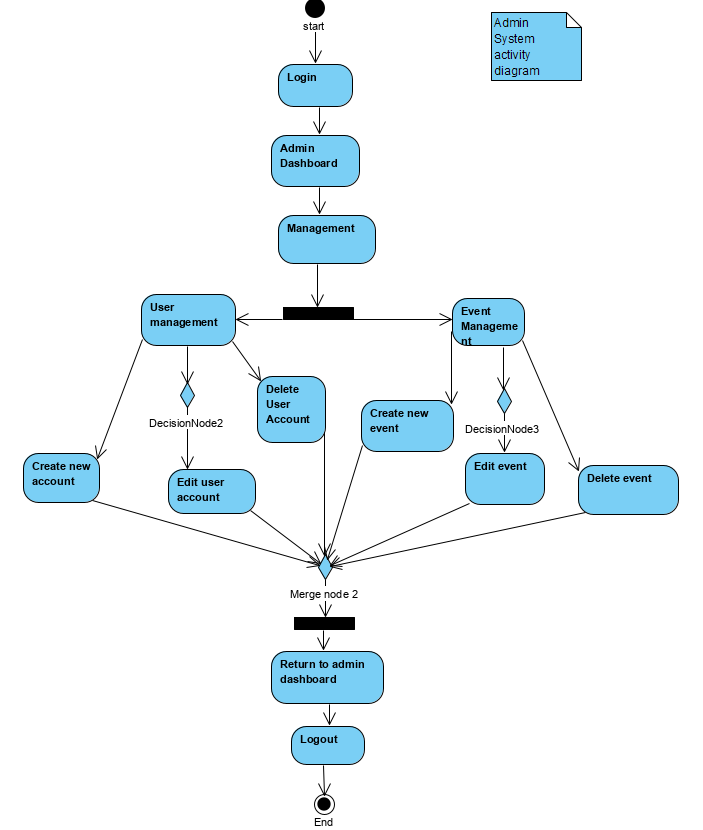
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Functional requirements | | | | |
| ID | Requirement | Priority | Owner | Source |
| 1.0 | Track campus events | High | Michael | Client brief |
| 1.0.1 | Bookmark campus events | Medium | Michael | Client brief |
| 1.1 | Track local events | High | Michael | Client brief |
| 1.1.1 | Bookmark local events | Medium | Michael | Client brief |
| 1.2 | Track points of interest | High | Michael | Client brief |
| 1.2.1 | Bookmark points of interest | Medium | Michael | Client brief |
| 1.3 | Track local support for students | High | Michael | Client brief |
| 1.3.1 | Bookmark local support for students | Medium | Michael | Client brief |

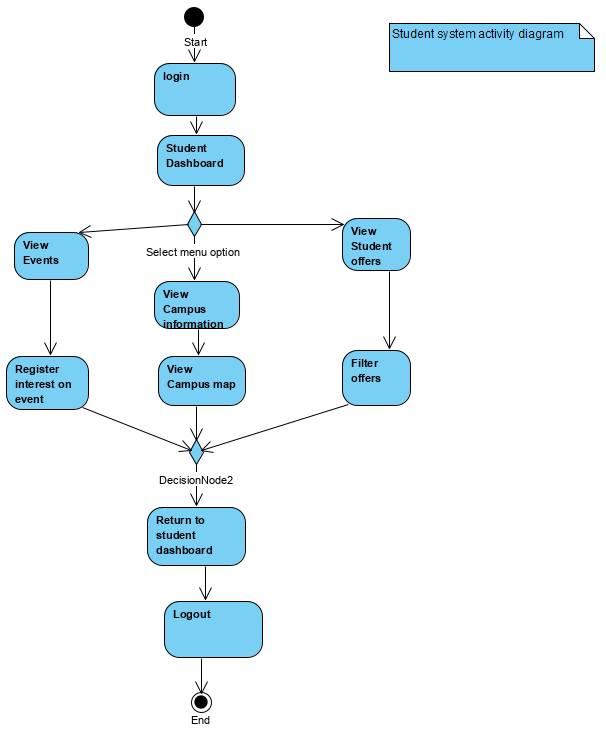
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Non-functional requirements | | | | |
| ID | Requirement | Priority | Owner | Source |
| 2. | Accessibility options | High | Michael | Client brief |
| 3. | Efficient data usage | High | Michael | Client brief |
| 4. | Simple to use | High | Michael | Client brief |
| 5. | Minimum device storage | Medium | Michael | Client brief |

# Use case diagram

****

# Activity diagram

****



# Class diagram

# Entity relationship diagram

# Skeleton Tables

# Entity Relationships

# Database Design

## Tables

Below are the tables that will be in the database. It has been normalised up to 3NF and only stores information that is necessary to save storage.

Campus\_events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Data type | Field size | Data Format | Key |
| Id | NUMBER | Dynamic |  | PRIMARY KEY |
| Name | VARCHAR2 | 200 |  |  |
| Start\_timestamp | TIMESTAMP |  | DD:MM:YYYY:HH24:MI:SS |  |
| End\_timestamp | TIMESTAMP |  | DD:MM:YYYY:HH24:MI:SS |  |
| Location | VARCHAR2 | 1000 |  |  |
| Details | VARCHAR2 | 4000 |  |  |

Local\_events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Data type | Field size | Data Format | Key |
| Id | NUMBER | Dynamic |  | PRIMARY KEY |
| Name | VARCHAR2 | 200 |  |  |
| Start\_timestamp | TIMESTAMP |  | DD:MM:YYYY:HH24:MI:SS |  |
| End\_timestamp | TIMESTAMP |  | DD:MM:YYYY:HH24:MI:SS |  |
| Location | VARCHAR2 | 1000 |  |  |
| Details | VARCHAR2 | 4000 |  |  |

Local\_support

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Data type | Field size | Data Format | Key |
| Name | VARCHAR2 | 200 |  | PRIMARY KEY |
| Location | VARCHAR2 | 1000 |  |  |
| Details | VARCHAR2 | 4000 |  |  |

Local\_poi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Data type | Field size | Data Format | Key |
| Name | VARCHAR2 | 200 |  | PRIMARY KEY |
| Location | VARCHAR2 | 1000 |  |  |
| Details | VARCHAR2 | 4000 |  |  |

All the tables above have a corresponding bookmark table that is used to link users with the data that they have bookmarked.

Bookmarked\_campus\_events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Data type | Field size | Data Format | Key |
| Username | NUMBER | Dynamic |  | PRIMARY KEY |
| Campus\_event\_id | NUMBER | Dynamic |  | FOREIGN KEY |

Bookmarked\_local\_events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Data type | Field size | Data Format | Key |
| Username | NUMBER | Dynamic |  | PRIMARY KEY |
| Local\_event\_id | NUMBER | Dynamic |  | FOREIGN KEY |

Bookmarked\_local\_support

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Data type | Field size | Data Format | Key |
| Username | NUMBER | Dynamic |  | PRIMARY KEY |
| Local\_support\_name | VARCHAR2 | 200 |  | FOREIGN KEY |

Bookmarked\_local\_poi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Data type | Field size | Data Format | Key |
| Username | NUMBER | Dynamic |  | PRIMARY KEY |
| Local\_poi\_name | VARCHAR2 | 200 |  | FOREIGN KEY |

## SQL

Below is the SQL code used in order to make the tables shown above.

|  |  |
| --- | --- |
| -- Make campus events table  CREATE TABLE campus\_events(  id NUMBER NOT NULL CONSTRAINT campus\_events\_id\_pk PRIMARY KEY,  name VARCHAR2(200) NOT NULL,  start\_timestamp TIMESTAMP NOT NULL,  end\_timestamp TIMESTAMP NOT NULL,  location VARCHAR2(1000) NOT NULL,  details VARCHAR2(4000) NOT NULL  );  -- Make sequence for trigger later  CREATE SEQUENCE campus\_events\_id\_sequence;  -- Trigger to do incerement  CREATE OR REPLACE TRIGGER campus\_events\_on\_insert  BEFORE INSERT ON campus\_events  FOR EACH ROW  BEGIN  SELECT campus\_events\_id\_sequence.nextval  INTO :new.id  FROM dual;  END;  -- Make bookmarked campus event table  CREATE TABLE bookmarked\_campus\_events(  username NUMBER NOT NULL CONSTRAINT student\_id\_bookmark\_campus\_events\_pk PRIMARY KEY,  campus\_event\_id NUMBER,  CONSTRAINT campus\_event\_id\_fk FOREIGN KEY (campus\_event\_id) REFERENCES campus\_events(id)  ); | -- Make local events table  CREATE TABLE local\_events(  id NUMBER NOT NULL CONSTRAINT local\_events\_id\_pk PRIMARY KEY,  name VARCHAR2(200) NOT NULL,  start\_timestamp TIMESTAMP NOT NULL,  end\_timestamp TIMESTAMP NOT NULL,  location VARCHAR2(1000) NOT NULL,  details VARCHAR2(4000) NOT NULL  );  -- Make sequence for trigger later  CREATE SEQUENCE local\_events\_id\_sequence;  -- Trigger to do incerement  CREATE OR REPLACE TRIGGER local\_events\_on\_insert  BEFORE INSERT ON local\_events  FOR EACH ROW  BEGIN  SELECT local\_events\_id\_sequence.nextval  INTO :new.id  FROM dual;  END;  -- Make bookmarked local event table  CREATE TABLE bookmarked\_local\_events(  username NUMBER NOT NULL CONSTRAINT student\_id\_bookmark\_local\_events\_pk PRIMARY KEY,  local\_event\_id NUMBER,  CONSTRAINT local\_event\_id\_fk FOREIGN KEY (local\_event\_id) REFERENCES local\_events(id)  ); |
| -- Make local support table  CREATE TABLE local\_support(  name VARCHAR2(200) NOT NULL CONSTRAINT support\_pk PRIMARY KEY,  location VARCHAR2(1000) NOT NULL,  details VARCHAR2(4000) NOT NULL  );  -- Make bookmarked support table  CREATE TABLE bookmarked\_support(  username NUMBER NOT NULL CONSTRAINT student\_id\_bookmark\_support\_pk PRIMARY KEY,  local\_support\_name VARCHAR2(200),  CONSTRAINT support\_name\_fk FOREIGN KEY (local\_support\_name) REFERENCES local\_support(name)  ); | -- Make points of interest table  CREATE TABLE local\_poi(  name VARCHAR2(200) NOT NULL CONSTRAINT local\_poi\_id\_pk PRIMARY KEY,  location VARCHAR2(1000) NOT NULL,  details VARCHAR2(4000) NOT NULL  );  -- Make bookmarked point of interest table  CREATE TABLE bookmarked\_poi(  username NUMBER NOT NULL CONSTRAINT student\_id\_bookmark\_poi\_pk PRIMARY KEY,  local\_poi\_name VARCHAR2(200),  CONSTRAINT poi\_name\_fk FOREIGN KEY (local\_poi\_name) REFERENCES local\_poi(name)  ); |

# User-interface design

The client has said that the system should be easy to use. In order to do this the design should be a simple and as straight forward as possible. The client should also work on multiple platforms and devices, so the front-end design should be designed around multiple devices instead of just one.

Home Page

The home page is designed to be as simple as possible. In the top left is a navigation menu button that will open the navigation menu on the left with buttons to all the different pages. The buttons to the pages are also on the main page, three wide and how ever many down, depending on the need. Having the buttons like this mean that no matter the size of the page they will be easy to see and use.

All other pages will link back to the homepage and the navigation bar will be accessible everywhere, so people should not get lost within the app.

The other pages will mainly be a list of buttons that when clicked will display more information and have a button to bookmark for later.

# Security

The program has a few different security levels that determine what can be seen. These levels are admin, event manager, lecturer/teacher, student and guest.

Having these levels of security greatly protects the software from unauthorised usage.

The admin will have overall control, meaning they can see and change everything. The event manager will be able to add and remove events. The lecturers will have the same control as students for now, but this can be changed depending on the future of the program. The students can see and bookmark the different things available and guests can only view things.

# Implementation of kiosk

# Innovations

# Test system

# Future evolutions

# Conclusion

# References

# Appendix