

# Software Design Specification

## Smart Student Attendance System *version 1.0*

---

# Table of Contents

<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 System Overview	1
1.3 Design Map	1
1.4 Definitions and Acronyms	1
<b>2. Design Considerations</b>	<b>1</b>
2.1 Assumptions	1
2.2 System Environment	1
2.3 Design Methodology	2
2.4 Risks and Volatile Areas	2
<b>3. Architecture</b>	<b>2</b>
3.1 Overview	2
3.2 Subsystem, Component, or Module 1 ...N	3
<b>4. Database Schema</b>	<b>3</b>
4.1 Tables, Fields and Relationships	3
4.1.1 Database Implementation	4
<b>5. High Level Design</b>	<b>4</b>
5.1 User Interface Modifications	5
5.2 Workflow sub-processes	8
<b>6. User Interface Design</b>	<b>9</b>
6.1 Application Controls	10
6.2 Screen 1... N	11
<b>Appendix: Project Timeline</b>	<b>11</b>

---

## Approved By

*Approvals should be obtained for project manager, and all developers working on the project.*

Name	Signature	Department	Date
Jordyn Brosemer		Electrical Engineering	12/16/2022
Bowang	YAN BOWANG	Electrical Engineering	12/16/2022
Duc Nguyen			
Yongqiang Zuo			

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to describe the design specifications taken into account when designing this application. This includes project planning, UI, and database/backend specifications. This app is designed to be a smart teacher attendance taking application. It is designed for any user to be able to mark attendance for weekly or single events as well as look at that data in the future. This app was designed in EE-192 Software Engineering and was designed to teach students process flow and agile methodologies in software development.

## **1.2 System Overview**

This app was designed to be used on an Android mobile device and all of its data is located on that device and not stored on any cloud. Users can log in to access their own data as this is a mobile device it should only be for one user so all phone data is shared among anyone who logs in on that phone.

All images you see on this page are not from a real application running on a real android phone. This app was designed, programmed and emulated using a pixel 6 phone using android studio on android api version 33.

## **1.3 Design Map**

## **1.4 Definitions and Acronyms**

**ID:** Identification Number

**UI:** User Interface

**SSAS:** Smart Student Attendance System

# **2. Design Considerations**

## **2.1 Assumptions**

- There would only be one user using the application because of the database will be designed locally in the phone
- The application will only be used by faculty administrators and teachers as the functionality for students and employees are not employed.

## **2.2 System Environment**

- The main operating system of the application would be Android.
  - The application is built and tested in Android Studio with Android Platform API 33 - Tiramisu.
-

## 2.3 Design Methodology

(Optional) - Summarize the approach that will be used to create and evolve the designs for this system. Cover any processes, conventions, policies, techniques, or other issues which will guide design work.

## 2.4 Risks and Volatile Areas

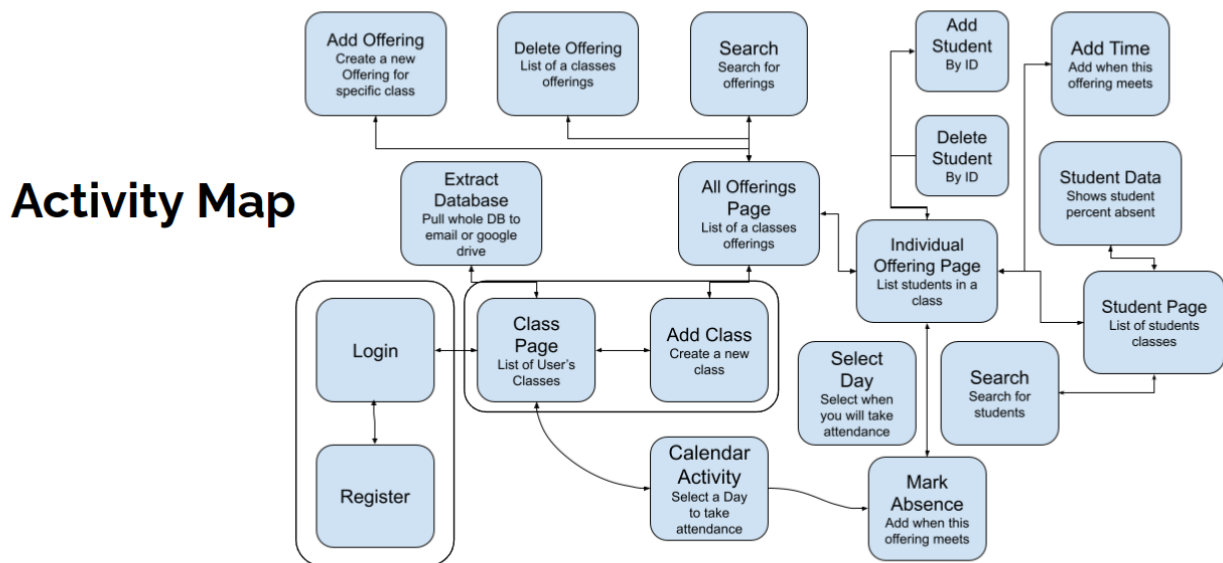
None have been identified.

## 3. Architecture

The architecture provides the top level design view of a system and provides a basis for more detailed design work

Provide or reference a detailed description and diagrams of the architecture..

Figure 1:



- Figure 1 is the detailed activity and application architecture of the smart student attendance system.

## 3.1 Overview

This section provides a high level overview of the structural and functional decomposition of the system. Focus on how and why the system was decomposed in a particular way rather than on details of the particular components. Include information on the major responsibilities and roles the system (or portions) must play.

Due to the requirements of each class containing class offerings and each offering containing students and attendance, this app performs like a nested list. Each list contains a list and that list contains multiple lists. Because this is difficult for humans, the app keeps track of the attachment between lists, such as a student's attendance in a class offering that is only listed on Sundays. The instructor just needs to add meeting times to a class as well as students and attendance from there is simple.

## 3.2 Subsystem, Component, or Module 1 ...N

*You only need to provide this level of detail for elements which are custom for this design. Do not go into gory detail. Goal is to get 80% of the elements figured out ahead of time.*

*Describe an element (subsystem, component, module, etc.) from architecture in further detail. When appropriate, include information on how the element is further broken down and the interactions and relationships between these subcomponents.*

- The key elements of designing the application is based on the functional requirement and non-functional requirements (Please see the system requirement documentation for more details).
- The main activity of the system would be:
  - Login Page
  - Course Page
  - Course Offering Page
  - Student Page
  - Database

## 4. Database Schema

### 4.1 Tables, Fields and Relationships

Required tables:

**Login Table:** Enables teachers to access their own individual mobile data and attendance platform.

- Teacher Username
- Teacher password

**Class Table:** Contains list of courses offerings for each class

- Course ID (Unique)
- Course Name
- Status
- Number of Offerings

**Class Offering Table:** All class offerings for each class. Contains a list of enrollment and list of meeting/scheduled times

- Course offering ID (Unique)
- Number of Students
- Course classroom

**Student Table:** All students. Each student contains their own attendance list and an enrollment list of class offerings they are in.

- Student ID (Unique)
  - Student Name
  - Student Email
  - Student Status
  - Student Payment Information
-

**Enrollment List:** Points students to offerings if they are enrolled.

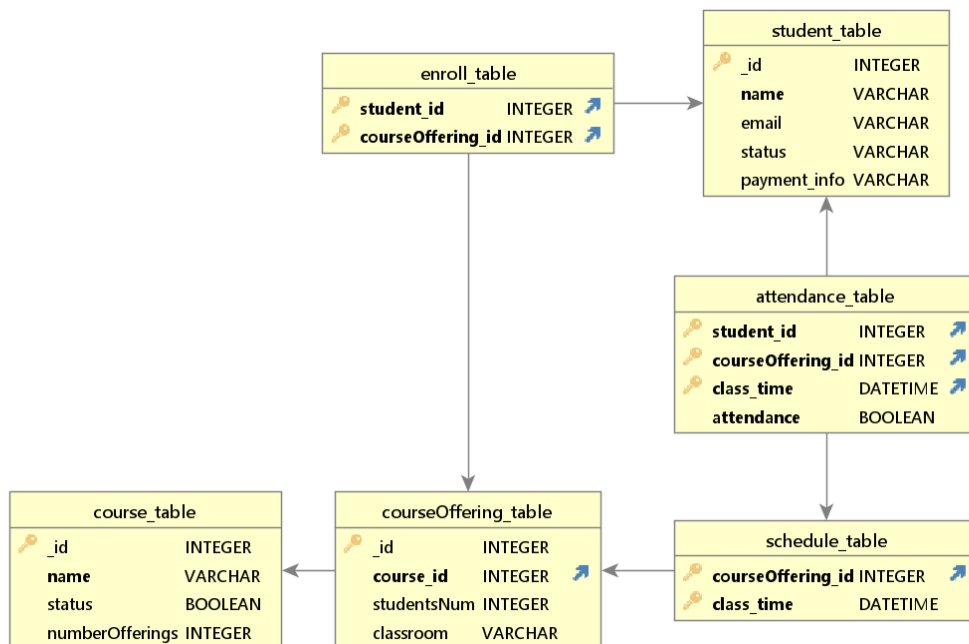
- Student ID (Unique)
- Offering ID (Unique)

**Attendance Table:** Points students to days their offerings are listed. Allows to take attendance.

- Student ID (Unique)
- Offering ID (Unique)
- Class\_time
- Present/Absent Status

#### 4.1.1 Database Implementation

Implemented these table requirements using a SQLite Database.

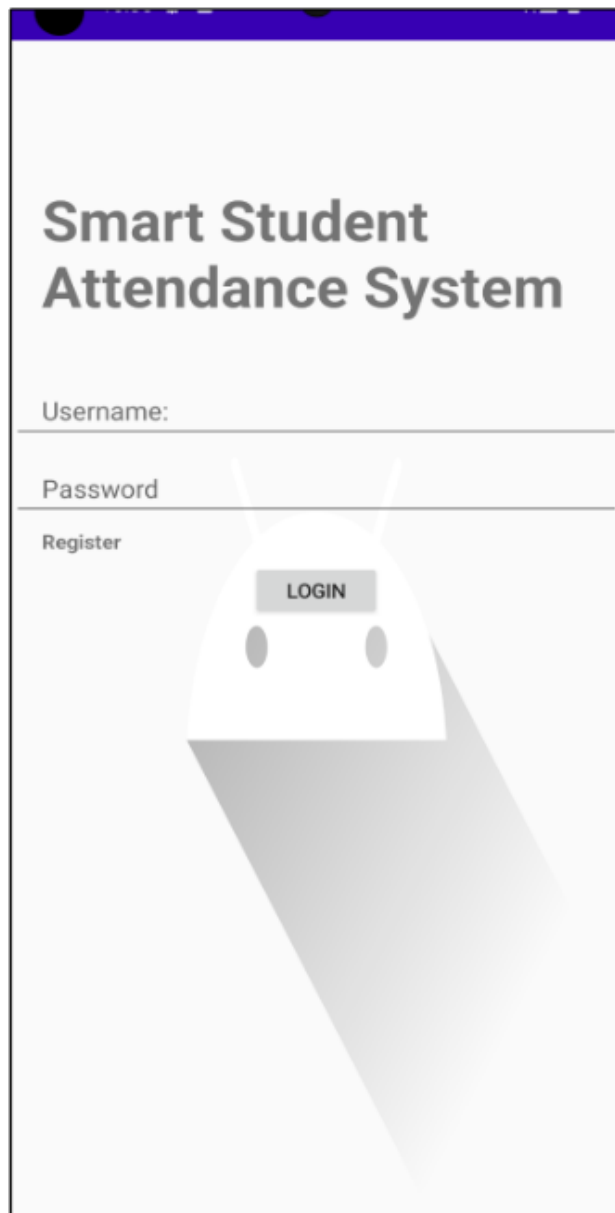


## 5. High Level Design

### 5.1 User Interface Modifications

- The high-level design would comprise of getting the user interacting with the key activities of the system
  - Login and SignUp Page
  - Course Page
  - Course Offering Page
  - Student Page

Login and SignUp Page:



Smart Student Attendance System

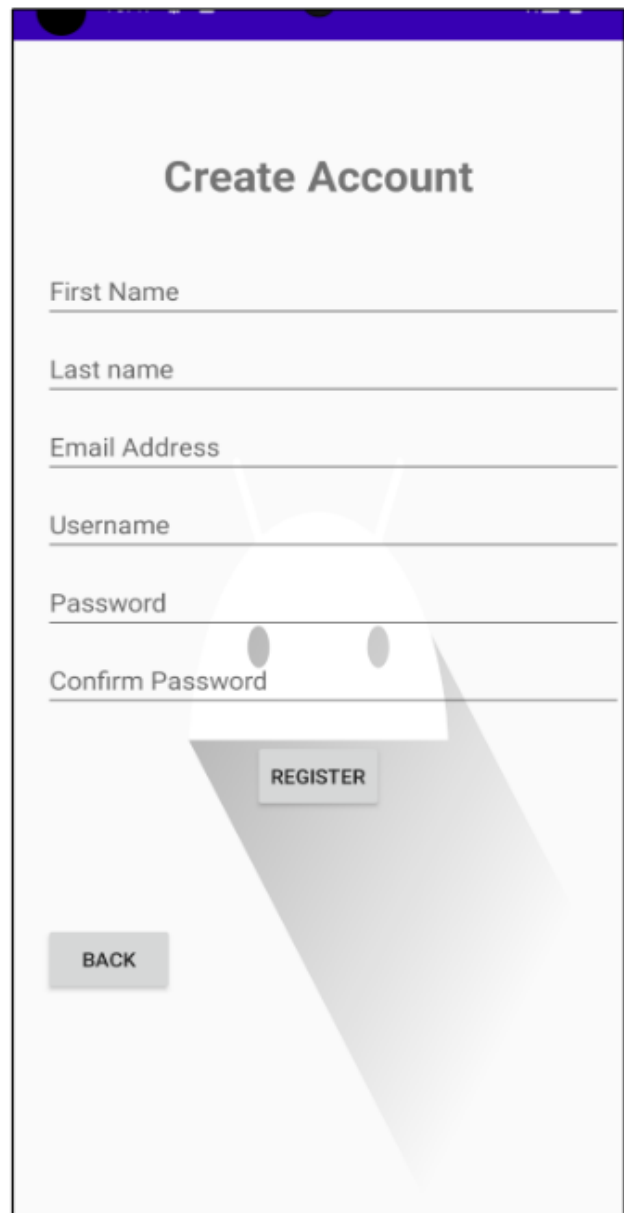
Username:

Password

Register

LOGIN

This is a mobile app interface for the 'Smart Student Attendance System'. It features a white Android robot character in the background. The interface includes a title, a 'Username:' label, a 'Password' label, a 'Register' button, and a 'LOGIN' button.



Create Account

First Name

Last name

Email Address

Username

Password

Confirm Password

REGISTER

BACK

This is a mobile app interface for the 'Create Account' page. It features a white Android robot character in the background. The interface includes a title, five input fields for 'First Name', 'Last name', 'Email Address', 'Username', and 'Password', a 'Confirm Password' field, a 'REGISTER' button, and a 'BACK' button.



Course Page:

[ADD CLASS](#) [GO TO CALENDAR VIEW](#) [SIGN OUT](#)

## Welcome Teacher

**Class ID:** 1

**Class Name:** Sociology

**# Offerings:** 2

---

[EXTRACT DATABASE](#)

## Create Class

Class Name

Class ID

Number of Class Offerings

[REGISTER](#)

[BACK](#)

Course Offering Page:

2:20

BACK

SEARCH

# CS111 OS

**Course Offering ID: 1**

Course ID: 1

Classroom: JCC101

Number of students: 40

**Course Offering ID: 2**

Course ID: 1

Classroom: JCC102

Number of students: 30

ADD OFFERING

DEL OFFERING

2:34

BACK

SEARCH

CS111 OS

**Class Offering ID: 1**

Max Student Num: 40

Student Name: Duc Nguyen

Student ID: 3

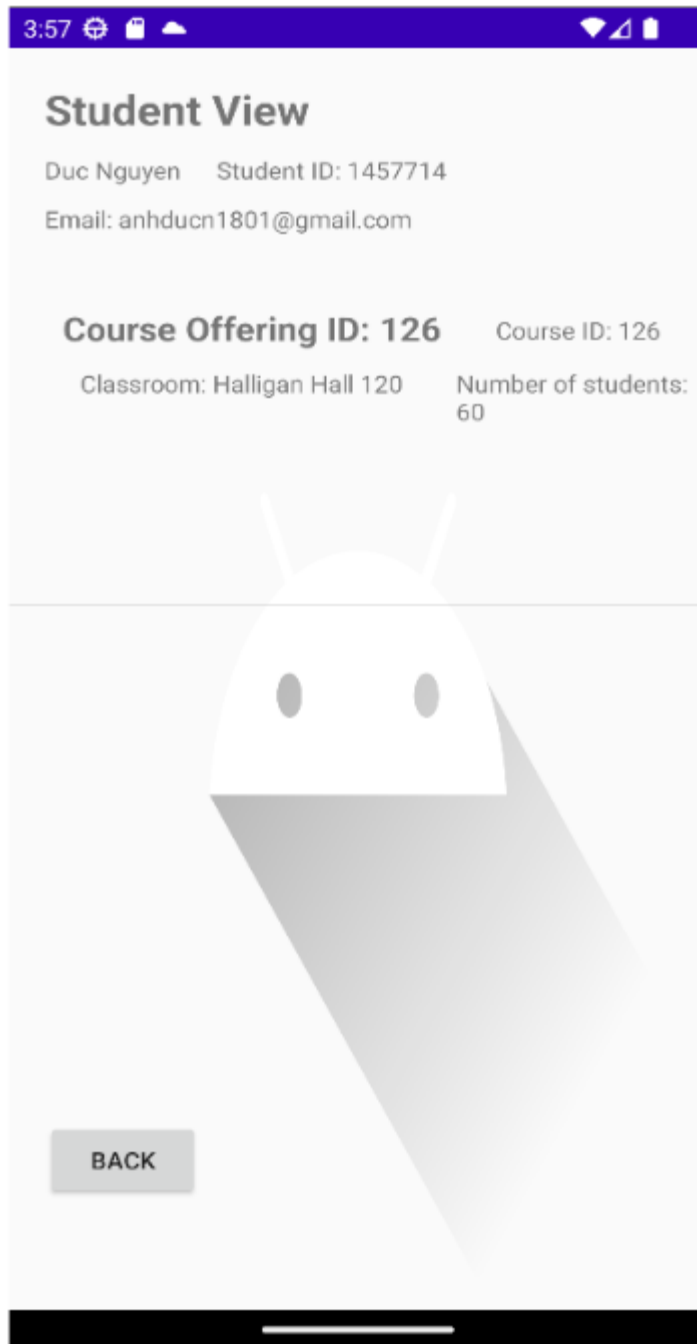
E-mail: dnguye22@tufts.edu

MARK ABSENCE

ADD STUDENT

DELETE STUDENT

Student Page:



## 5.2 Workflow sub-processes

- The workflow sub-processes will be divided among the members to work on a main activity and their main functionality with one member working specifically on the SQLite Database.
- Further sub-processes activity will be discussed in the next section.
  - Jordyn : Login Page
  - Bowang : Course Offering Page
  - Duc : Student Page

- Yongqiang : Database

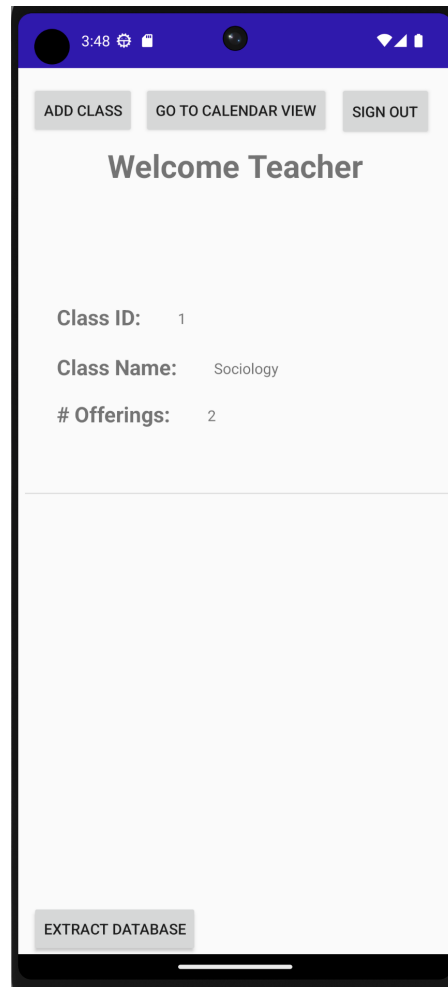
## 6. User Interface Design

*This section provides user interface design descriptions that directly support construction of user interface screens.*

- More attributes of the activity's entities such as students, courses, and course offerings are listed in the Software Requirement Specification.
  - This section will discuss about the functionality of each main page and the low level design of each page
    - Login and SignUpPage:
      - The user will be able to create a user
      - The user cannot create an account unless all the required information are entered
      - The user cannot create an account if the the confirm password does not match the password
      - The user will be able to login to the application
      - The user cannot login if the username and password does not match the ones being registered in the database
      - There would only one user using the phone so linking extra user on the database table are not implemented
    - Course Page:
      - The user will be able to create/delete a course
      - The user can view all the course using a calendar view
      - The user needs to be able to extract the database
      - The user will be able to view the course information, offerings, and data by clicking on the items.
      - Sub-system: Calendar View
        - The calendar view will show all the course offerings for the date using a calendar UI.
        - The user will be able to go on the date assigned and mark absence and update the database of the students
    - Course Offering Page:
      - The user will be able to create/delete a course offering
      - The user can view the course offering information
      - The user will be able to create/delete a student with an option to click to view the data of the students
      - The user can mark absence by searching for the dates the being created
      - The user can search student by student ID
      - Sub-system: Mark Absence
        - A table of students of the date that can be marked for attendance
      - Sub-system: Create Time
        - Create a time schedule for the course offerings so that the users can mark attendance.
    - Student Page:
      - The user can see all the information of the student which include the student ID, status, email, and payment method
      - The user can see all the course offerings that the students are enrolled in
      - The user can check the attendance percentage of that students for the selected course offerings
-

- The user can view the date that the students attended/missed in the classes.

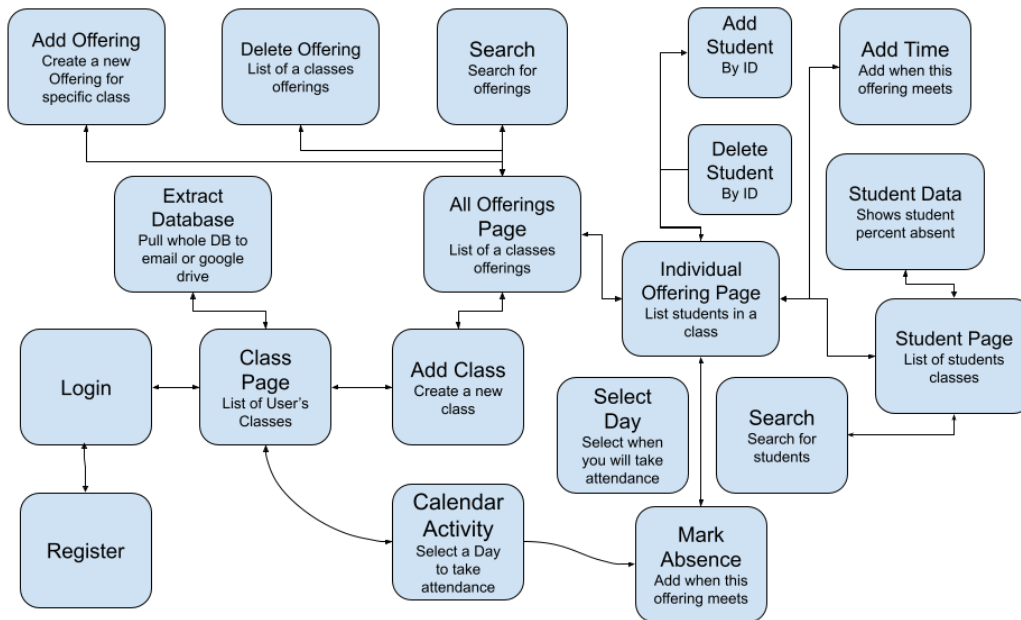
## 6.1 Application Controls



All screens will contain a back button to return to where the user has come from as well as a title banner that gives the user an idea of where they are. All buttons will have a large font or indication that clicking will do something so the user knows exactly where they will be going upon interaction.

Any page with scrollable lists will have a scroll bar on the far right to see how long the list as well as flash upon being clicked so the user knows that they have clicked or that the list is clickable. Any information on what pages do what should be found in the user manual.

## 6.2 Screen 1... N



## Appendix A: Project Timeline

*Reference the Microsoft project Binder Request Release 2 – Development..*

Nov 1 - Nov 15: Come up with functional requirements and nonfunctional requirements.

Nov 16 - Nov 26: Integrate all the requirements.

Nov 27 - Dec 4: Database design.

Dec 5 - Dec 10: Individual pages design and coding.

Dec 11 - Dec 12: Pages integration.

Dec 13 - Dec 16: Final report.