Software Design Document for Scholarli

College of Engineering and Computer Science

Florida Atlantic University

CEN4010 – Principles of Software Engineering

Dr. Safak Kayikci

Prepared By: Juan Rodriguez, Kevin Tran, Thau Tran-Nguyen

Revisions Page

Name	Date	Version
Thau Tran-Nguyen	7/11/2024	1.0
Juan Rodriguez	7/11/2024	1.0
Kevin Tran	7/11/2024	1.0

1. Introduction

1.1 Purpose

The purpose of this Software Design Document (SDD) is to provide and outline the architecture and system design for Scholarli development, which is a semester-based student assignment tracker. This document serves as a guide for the development team, users and investors involved in the project.

1.2 Scope

Scholarli is aimed towards students in colleges, universities or high school. This project will efficiently organize several types of assignments over the semester. It has a function that will leave notification reminders for students to study or start an assignment. Our aim is to provide these students with a fully customizable experience depending on how much they want Scholarli to aid them, students can minimize involvement to only notify students once or twice about a certain assignment or to notify them multiple times based on their lifestyle. The main objective is to give students a minimal maintenance assignment tracking resource that can be flexible and customizable.

1.3 Overview

This document provides an overview of Scholarli's system architecture, data design, component design and human interface design. This can be used for the development and investors to understand the technical design behind Scholarli. Users can use this document in the future to help understand and communicate changes with the development team.

1.4 Reference Material

https://github.com/KatherineSantore/AssignmentTracker_Client

https://www.researchgate.net/figure/High-level-software-architecture-of-DSMS_fig5_332656904

1.5 Definitions and Acronyms

API: Application Programming Interface

OS: Operating System

HTTPS: Hypertext Transfer Protocol Secure

Celery task scheduler: A tool used in software development to schedule and manage periodic tasks, ensuring they are performed at the correct times.

SendGrid email API: An API provided by SendGrid for sending emails from applications in a reliable and scalable manner.

GUI: Graphical User Interface

IDE: Integrated Desktop Environment

SDD: Software Design Document

ER: Entity Relationship Diagram

PDF: Portable Document Format

FAQs: Frequently Asked Questions

2. System Overview

This app is designed to be a cross-platform student assignment tracker aimed to help students. It provides a seamless user experience across multiple devices. It offers features like assignment creation, synchronization and reminder notifications that are semester based for students to easily track their time in college or university. The app consists of a frontend, which includes the user interface for Android, iOS and desktop. The backend consists of an authentication service, assignment management service, synchronization service, notification service, and a database. Shown below is a system overview of Scholarli.

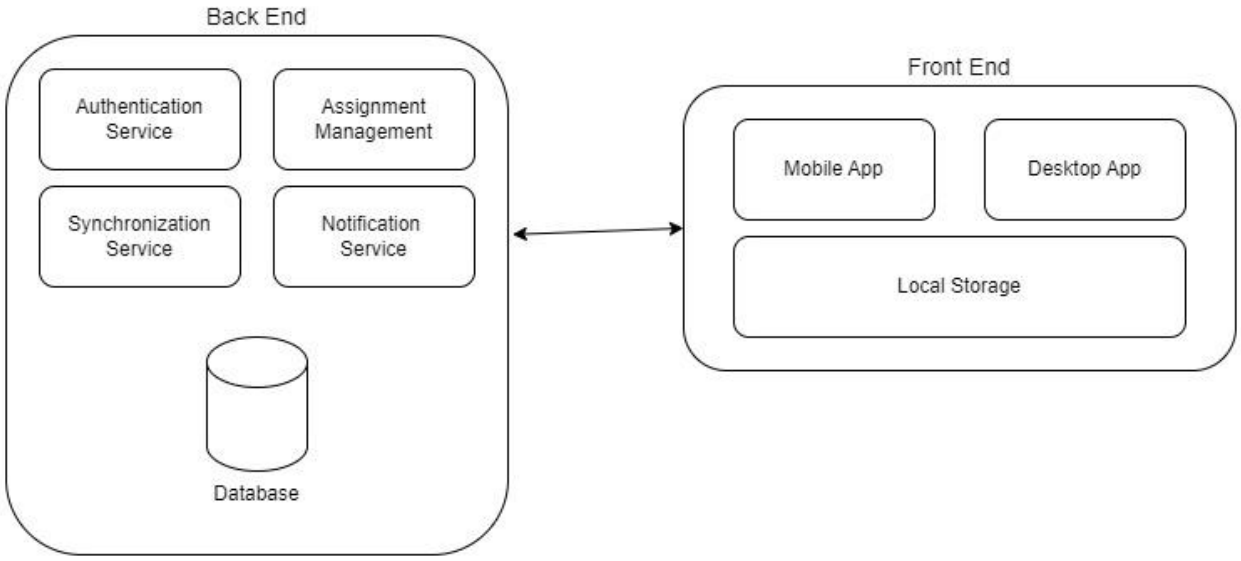

Figure 1: System Overview

3. System Architecture

3.1 Architectural Design

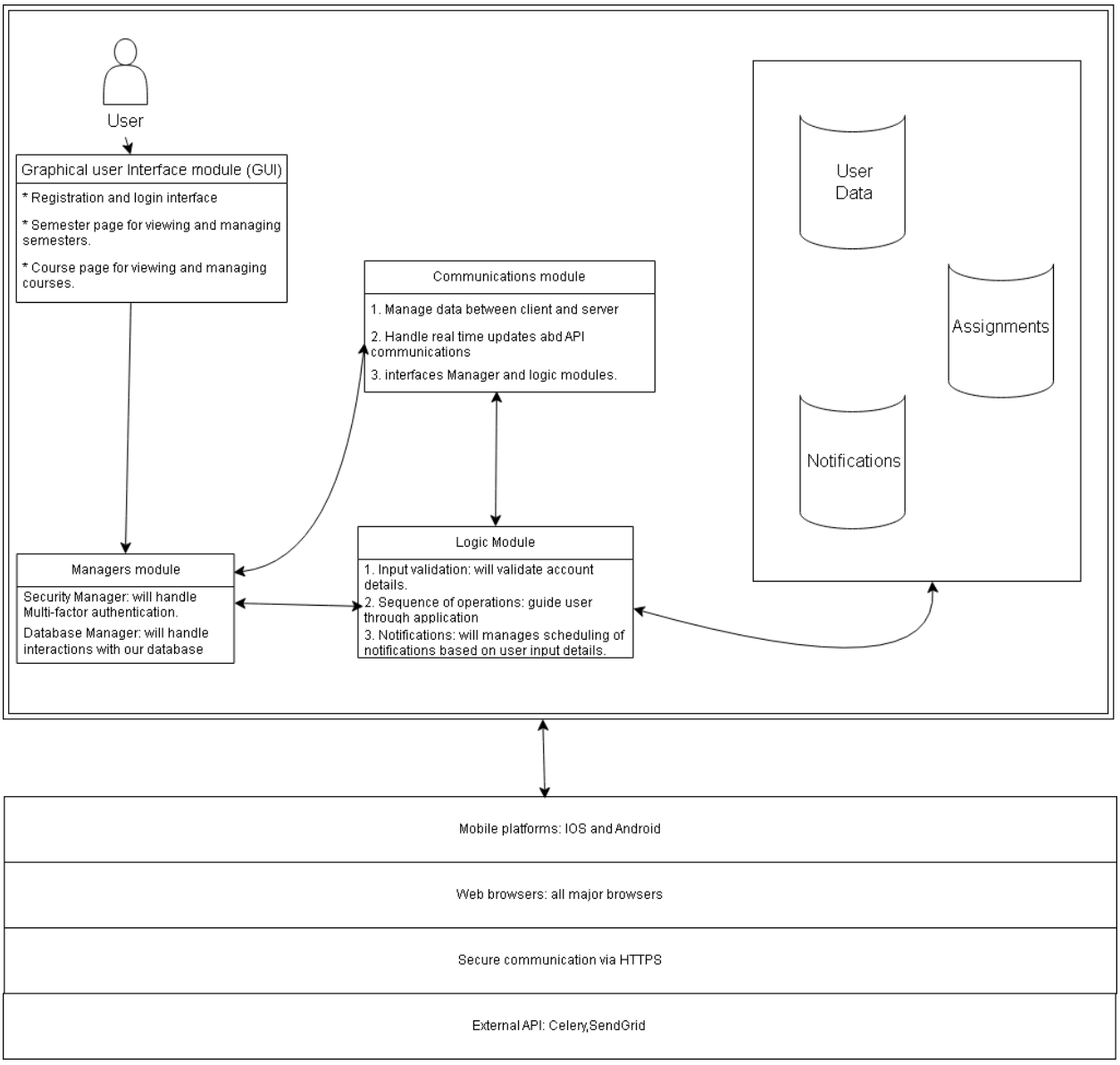

Figure 2: Architectural Design Diagram

Above is the high-level system design diagram for Scholari, this diagram details the high-level components for our program and small details on their functions.

3.2 Decomposition Description

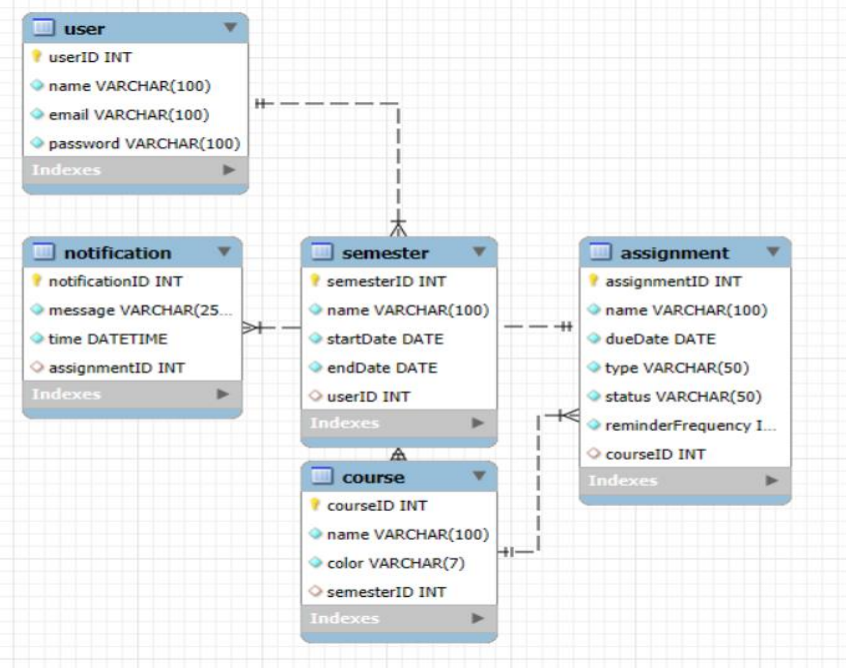

Figure 3: ER Diagram

The entity relationship diagram for Scholarli illustrates the connected components needed for the system. Users, represented by the “User” table, manage their academic profiles and schedules. These are linked to the “Semester” table, which stores data about individual academic periods such as start and end dates of a semester. The “Course” table contains information on specific subjects within each semester. Assignments are managed through the “Assignment” table, which helps with tracking details such as due dates and statuses of assignments. Notifications are handled by the “Notification” table, ensuring timely reminders for each assignment. Through these relationships of each table, Scholarli makes academic management simple and aids students in being organized and on top of their coursework for the semester.

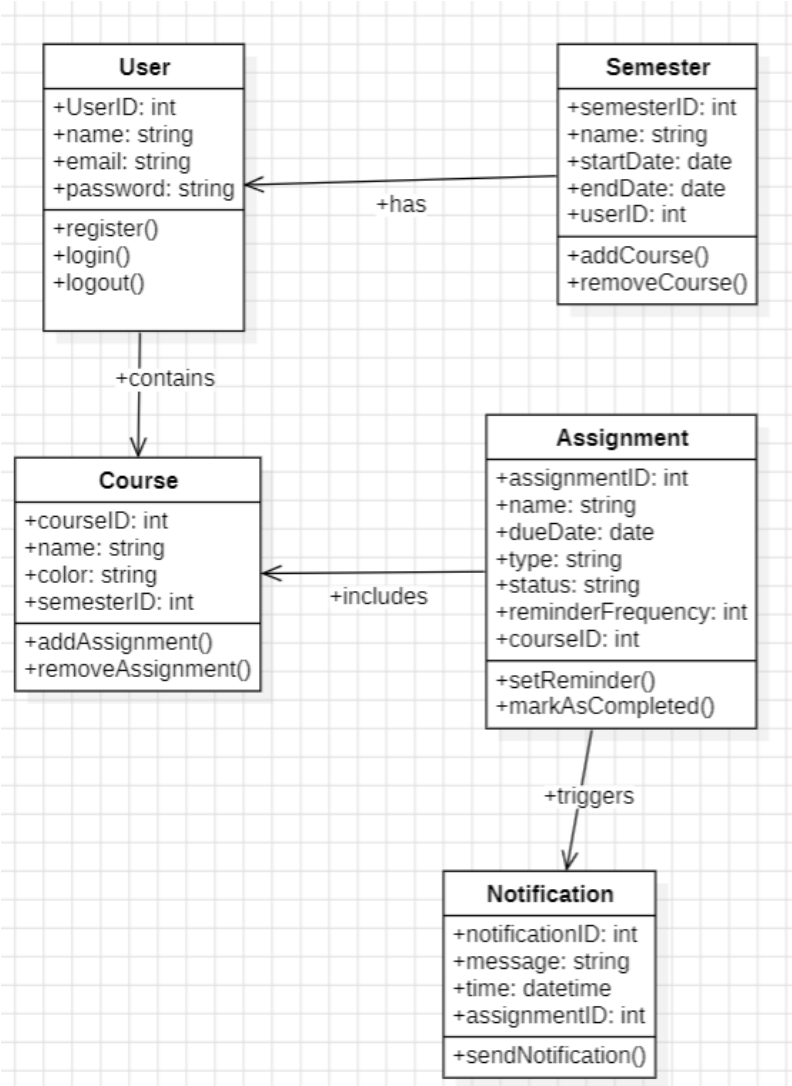

Figure 4: Object Class Diagram

The Scholarli system is architecturally divided into modular subsystems, each with their own responsibilities: User Management for account handling, Semester Management for organizing academic periods of time, Course Management for managing courses within semesters, Assignment Management for tracking specific assignments for a class, and the Notification system for sending out reminders on assignments. These subsystems interact effortlessly, with user data flowing from the User Management system to the Semester Management system, which helps organize academic periods. This data then informs the Course Management system, which includes multiple assignments tracked by the Assignment Management system. The Notification System keeps users informed about their assignments, creating a very comprehensive and connected framework for efficient academic assignment planning.

4. Data Design

4.1 Data Description

The data in Scholarli is handled within a single database, organized into separate tables. These tables include user profiles and settings, semester and course data, assignment information, and notifications. The database is backed up on a regular basis to make sure data integrity and availability is intact in the event of disk failures.

4.2 Data Dictionary

API:

- Type: Interface
- Description: Facilitates communication between the front-end and back-end systems, ensuring seamless data exchange.

Apps:

- Type: Interface
- Description: Mobile and web applications for students to use on multiple platforms, providing access to assignment tracking, course management, and notification settings.

Backups:

- Type: Data Storage
- Description: Ensures that important data is not lost in the event of a storage device failing, such as a dead hard drive.

User Profile and Settings Table:

- Type: Data Storage
- Description: Stores user profiles, account settings, and preferences for the user.
- Attributes: userID, name, email, password, preferences

Semester and Course Table:

- Type: Data Storage
- Description: Contains details about assignments.
- Attributes (Semester): semesterID, name, startDate, endDate, userID
- Attributes (Course): courseID, name, color, semesterID

Assignment Table:

- Type: Data Storage
- Description: Contains details about assignments.
- Attributes: assignmentID, name, dueDate, type, status, reminderFrequency, courseID

Notification Table:

- Data Storage
- Description: Stores notifications for assignments
- Attirubutes: notificationID, message, time, assignmentID

User Profile:

- Type: Object
- Description: Represents a user's profiles, containing attributes such as userID, username, email, password, and preferences.

Semester:

- Type: Object
- Description: Represents an academic semester, that contains attributes such as semesterID, name, start date, end date, and userID.

Course:

- Type: Object
- Description: Represents the courses within a semester, contraining attribute such as courseID, name, color, and semesterID.

Assignment:

- Type: Object
- Description: Represents an assignment, containing attributes such as assignmentID, name, due, date, type, status, reminder frequency, and the coursseID.

Notification:

- Type: Object
- Description: Represents a notification for an assignment, which contains attributes such as the notificationID, message, time, and assignmentID.

5. Component Design

User Authentication: This is responsible for user registration, login, and session management. It ensures that only registered users can access the app.

Assignment Management: This allows students to add, view, edit, and delete assignments. It should also allow setting priorities and due dates for assignments.

Notification: This sends reminders to students about upcoming assignments. It could use local notifications.

User Interface: This includes all the pages and visual elements of the app. It will be user-friendly.

Database: This is responsible for storing and retrieving data. It could use a local database on the device for offline access, and sync with a cloud database when online.

Synchronization: This would support would handle syncing data across devices.

Settings: This allows users to customize the app according to their preferences

Help and Support: There will be a FAQs page and a contact form for the development team. This is where users can report bugs or suggest new features.

6. Human Interface Design

6.1 Overview of User Interface

Logical characteristics:

- Registration and Login: When users open the app, there will be a registration and log in page. It will display two textboxes for the user to enter the log in information, which is their email address and password. If the user does not have an account, there will be another button for a user to register.
- Semester Page, View Semester: Upon logging in, a page displays all user-added semesters. Users can add semesters by clicking the plus on the top right of the page. If the user wish to delete a semester, they can swipe left on the semester. That will prompt the user to confirm the deletion by typing "CONFIRM" in the textbox. Clicking the plus button to add a semester will pop-up a screen to name the semester and set the duration of the semester.
- Display All Courses, Add/Remove Courses: By clicking a semester, the user will be greeted by a courses page. This page will display all the user added courses for that semester. Users can add a course by clicking the plus button on the top right corner of the page. Similar to semester deletion, users can delete a course by swiping left and confirming. Upon clicking "Display All Courses," user will be greeted with a half-page calendar of all the courses color coded (which is chosen by the user). Below the calendar will be the upcoming deadlines of assignments for all courses. There will be an option to convert the page to a PDF format for printing.
- Adding a Course: When users add a course, there will be a pop-up that allows the user to name the course and set a color for the course. More options will be added once there are feedback and demand for more features.
- Display One Course: When users click on one course, the calendar for that course will occupy half the page and the upcoming assignments. There will be an option to convert the page to a PDF format for printing.
- Add Assignments: To access add assignments, users will need to navigate to a course page. From this page, users can click the add assignments button on the top right corner of the page. This will pop-up a page for users to add the assignment name and type. The form will adjust accordingly to the type that the users choose. Options of assignments include homework, quiz, exam, reading, laboratory, essay and more upon users' feedback and requests.
- Edit Assignments: Users have the option to edit assignments once created. They can change all aspects of an assignment. From this page, they are also able to set the status of an assignment from "Not Started, In Progress, Completed."
- Notifications: Users will receive constant notifications. For several types of assignments, users will get notifications to start an assignment or to study for an upcoming quiz or exam within the appropriate timeframe.

Optimization: The application will be optimized to ensure rapid loading times. This is an application intended to be used on mobile devices, so allowing rapid access is crucial for effective use. We will aim to minimize use of resources and use efficient data storing techniques.

6.2 Screen Images

Shown below are the sample screen images for Scholarli. It shows the various pages that the user will be able to interact with.

Figure 5.1: Login Page

This is the landing page when the user opens the app. It will display the logo, name and will prompt the user to login. The user can enter their email into the email field and their password in the password field, respectively. If the user's email and password is wrong upon the user pressing the login, then the text field will animate shake and red text will tell the user that the email or password is wrong. There is also a button for register, which will lead users to a simple registration page.

Figure 5.2: Semester Page

This is the semesters page that will display the users previously entered semesters. The page layout will include a top header stating "semesters" followed by sectioned horizontal labels with the user entered name for the semester. The page will also include a "+" at the top right corner of the screen to allow the user to add new semesters. To delete any semesters, the user will need to swipe to the right on the semester that the user wishes to remove.

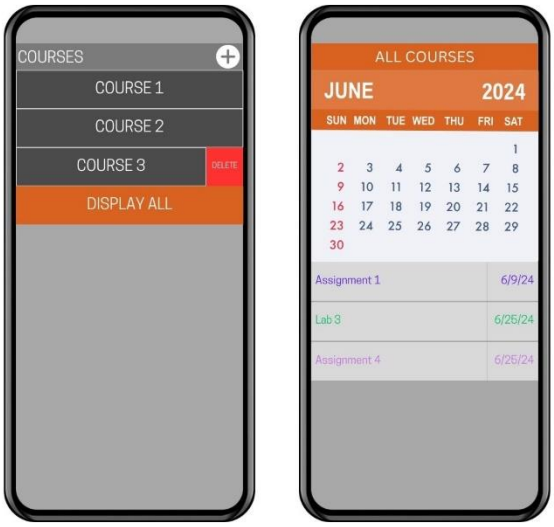

Figure 5.3: Course Page and Course Calendar

The image on the left displays an interaction that is possible on our course screen. Once inside the course page we can interact with the user entered courses. Users are also able to add any course using the "+" in the top right corner. The user is also able to delete any course by swiping right on the desired course. Also on this screen is a display all button, which will display all courses in this semester on our calendar screen. Users can also opt to display one course at a time by simply tapping on that course. The image pictured on the right is our calendar view when the display all button is interacted with. We see the calendar along with the upcoming events listed below. The colors seen are set by the user when the course is created/edited to distinguish the courses visually.

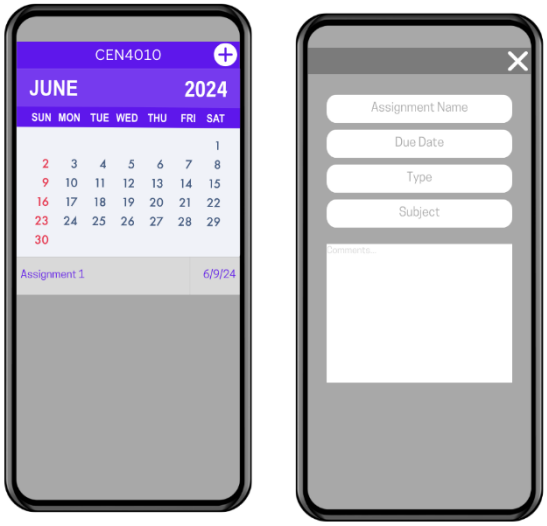

Figure 5.4: One Course Page and Assignment Creation Page

On the left, we see our calendar screen displaying only one course, the color running along the top banner is the set color used to distinguish events for this course. On this screen we also see a "+" this is used to add another course to your Calander view. To add any events users must tap on the date that they wish to add any event to. On the right we see the screen that is displayed when a user attempts to add an event. This screen allows users to enter event name, due date, event type, subject, and a small description for this event.

6.3 Screen Objects and Actions

Scholarli has a variety of screen objects and actions to help the user efficiently navigate the app. They are listed below with a brief description. In the Scholarli app, there will various text fields as the user navigates the app. Each text field serves a different purpose.

- Email: This is where the user will enter the email address that they used to register their account on the app when they login.
- Password: This is where the user will enter the password that they used to register their account on the app when they login.
- Assignment Name: When creating an assignment, this is where the user can enter the assignment name.
- Due Date: When creating an assignment, this is where the user will enter the due date of the assignment.
- Type: When creating an assignment, the user can start typing the assignment type and a preset list of assignments will appear for the user to auto complete or they can continue typing to create a new assignment type.
- Subject: When creating an assignment, the user can start typing the assignment subject, which will pop up a preset list of their added courses.
- Comments: When creating an assignment, the user can add comments to the assignment. This field is optional, and the assignment will be created without requiring this field to be filled.

In the Scholarli app, there will be different buttons the user can interact with as they navigate the app.

- Login: The user will press this button upon login. If the user enters the wrong credentials, then this button will vibrate slightly after the user pressed it.
- Register: This button will lead the user to a registration page, where they can enter their name, email address and password.
- Add (“+”): On different creation screens (course creation, semester creation, assignment creation), there will be an add button. This is where the user can start the creation process.
- Exit (“x”): Similarly, the user can cancel the creation by clicking on the exit button on presented different screens.
- Semester/Course: Upon clicking a semester, course or assignment, more details can be viewed by the user.

Scholarli will have a calendar object for the user to interact with. Listed below are the different actions the user can take.

- Click on date: When the user clicks on the date, the bottom half will display the assignment due that day. If there are no assignments, it will display “No assignments due today”

Another important action the user can do in Scholarli is to swipe. These are the actions the user can do when swiping.

- Remove semester/course: The user can swipe left on a semester or course to remove it. When swiping left, a red “DELETE” button will appear, and it will allow the user to remove the semester or course.
- Traverse calendar months: The user can swipe left and right to traverse the calendar months, and the bottom half of the page will reflect each month.

7. Requirements Matrix

Requirement	Description	Design Component
2.2.1	Functional: Validity check on input	Login/Sign-up user forms
2.2.2.1	Functional: Exact sequence of operations – Semesters	Semester creation, editing and deletion component
2.2.2.2	Functional: Exact sequence of operations – Calendar View	Calendar view component
2.2.2.3	Functional: Exact sequence of operations – Assignments	Assignment creation, editing and deletion component
2.2.3.1	Functional: Response to abnormal situations - Overflow	Figma for user interface design and SQLite for database management
2.2.3.2	Functional: Response to abnormal situations – Communication Facilities	Network libraries for android, iOS and web
2.2.3.3	Functional: Response to abnormal situations – Error Handling and Recovery	Sentry, Crashlytics, or Bugsnag provide real-time error tracking and SLF4J for Android
2.2.4	Functional: Effect of parameters	Database managers and various libraries can be used to fulfil task entry and due dates customization
2.2.5	Functional: Relationship of outputs to inputs - Notifications	Notification settings, push notifications, Preset formula for smart notifications for assignments.
2.2.6	Functional: Cross-Platform Compatibility	Responsive design, cross-platform framework
2.3.1	Non-functional: Reliability	Uptime and availability metrics
2.3.2	Non-functional: Availability	Uptime and availability metrics
2.3.3	Non-functional: Security	Multi-factor authentication via push notifications
2.3.4	Non-functional: Maintainability	User submitted bug reports and documentation
2.3.5	Non-functional: Portability	Kotlin multiplatform can be used to port share code between different platforms like Android, iOS, web and desktop.
2.3.6	Non-functional: Performance	Application performance monitoring tools