
Software Requirements Specification and Design document

for

CSUSM Student App

Version 1.0 approved

Prepared by JJ Javier, Kenneth Wang, William Phong

JKW

5/19/2023

Table of Contents

Table of Contents	ii
Revision History	iii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 About our project team	1
1.4 Intended Audience and Reading Suggestions	1
1.5 Product Scope	2
2. Statement of Work	2
2.1 Communication	2
2.2 Dependencies and Constraints	2
2.3 Design, Development, and Implementation Methods	2
2.4 Change Management	3
3. Timeline	3
4. Overall Description	4
4.1 Product Perspective	4
4.2 Product Functions	4
4.3 Technical requirements	5
4.4 Operating Environment	5
4.5 Design and Implementation Constraints	5
4.6 User Documentation	5
4.7 Assumptions and Dependencies	5
5. External Interface Requirements	5
5.1 User Interfaces	5
5.2 Hardware Interfaces	5
5.3 Software Interfaces	5
5.4 Communications Interfaces	5
6. System Features / Functional Requirements	6
6.1 Login	6
6.2 Create User	6
6.3 Calendar	7
6.4 Parking Availability Display	8
7. Nonfunctional Requirements	8
7.1 Performance Requirements	8
7.2 Safety Requirements	9
7.3 Security Requirements	9
7.4 Software Quality Attributes	9
7.5 Business Rules	9
8. System Architecture	10
8.1 Architectural Design	10
8.2 Decomposition Design	14
8.3 Design Rationale	18
9. Data Design (only applicable if your project has a database component)	19
9.1 Entity relationship design	19
9.2 Data dictionary	19
10. Other Requirements	20

Appendix A: Screen Images	20
11. Github Link	24
12. Evolution	24
12. Workflow Diagram	25
13. User Manual	26

Revision History

Name	Date	Reason For Changes	Version
JJ J, Kenneth W, William P	3/29/2023	Initial Draft	1.0
JJ J, Kenneth W, William P	4/2/2023	Final Draft	1.0
JJ J, Kenneth W, William P	5/19/2023	Final	1.0

1. Introduction

1.1 Purpose

The purpose is to create a mobile application that provides a streamlined way to access school related information. The application will provide students of CSUSM a way to monitor their school assignments and be delivered relevant information catered to their needs. Students and professors will be able to create accounts and receive notifications and other information related to their classes. This SRS will describe all of the software requirements and functions for the CSUSM Student Application. The document will describe necessary requirements and functions for the application, and provide visual representations through diagrams to allow for further understanding of the system.

1.2 Document Conventions

This document follows IEEE formatting conventions. The document is written in Arial, double spaced, and font sizes for main headers are 16 pt, and subsection headers and body sections are 12pt. Section headers and subsections are numbered and bolded. Every requirement statement has its own priority.

1.3 About our project team

William Phong CSUSM ID: 201347983, email: phong005@csusm.edu

JJ Javier CSUSM ID: 200790153, email: javie032@csusm.edu

Kenneth Wang CSUSM ID: 201031719, email: wang175@csusm.edu

JKW is a 3 man startup group based in California focused on improving student's day to day lives. We strive to provide and provide solutions that adapt to an individual's personal education needs by creating applications that help students use their time efficiently and provide creativity in balancing their everyday needs.

1.4 Intended Audience and Reading Suggestions

The intended audience of our proposal document are students of our SE370 class, the professor, administrators, and mentors of any kind. The SRS will also be utilized by the developers throughout the project. This SRS contains information pertaining to the top level requirements of the project, statement of work, timeline, and specific features of the product. The suggested reading order of this document is in order, with 6 main sections, and subsections underneath each one.

1.5 Product Scope

The purpose of this application is to provide a streamlined process to enhance students' day to day lives. An inefficient or bloated user interface leads to less usage of the software and ends up not providing any benefit to students. We want to optimize the user experience towards convenient and efficient purposes, allowing students to upbring their success rates while keeping track of their progress. This product aims to improve students' time management and prioritize each individual's personal needs.

2. Statement of Work

The JKW Group and CSUSM will work and communicate together in order to provide a product that meets the requirements provided.

2.1 Communication

Our team has come up with multiple solutions in order to maintain effective communication throughout our semester long project. In the case of any urgent messages such as difficulties/misunderstandings, changes or updates, clarifications, etc, we have a team Discord server as well as a messenger group chat. To address more specific and significant occurrences, the team will have weekly in person meetings in order to set our sights clear and make sure we stay on track to the schedule we commit to. We are also utilizing Github and Monday to update each other on our progress. The team, if necessary, will be able to text or email each other as well as the professor for any unclear concerns while also having each other's outlines and objectives for what we expect from each other.

2.2 Dependencies and Constraints

We will be utilizing Java for the frontend of the mobile application as well as implementing the backend. A MySQL database and connector will be used to establish a connection between the application and database. Dependencies that is used for this application is androidx and mysql connector.

2.3 Design, Development, and Implementation Methods

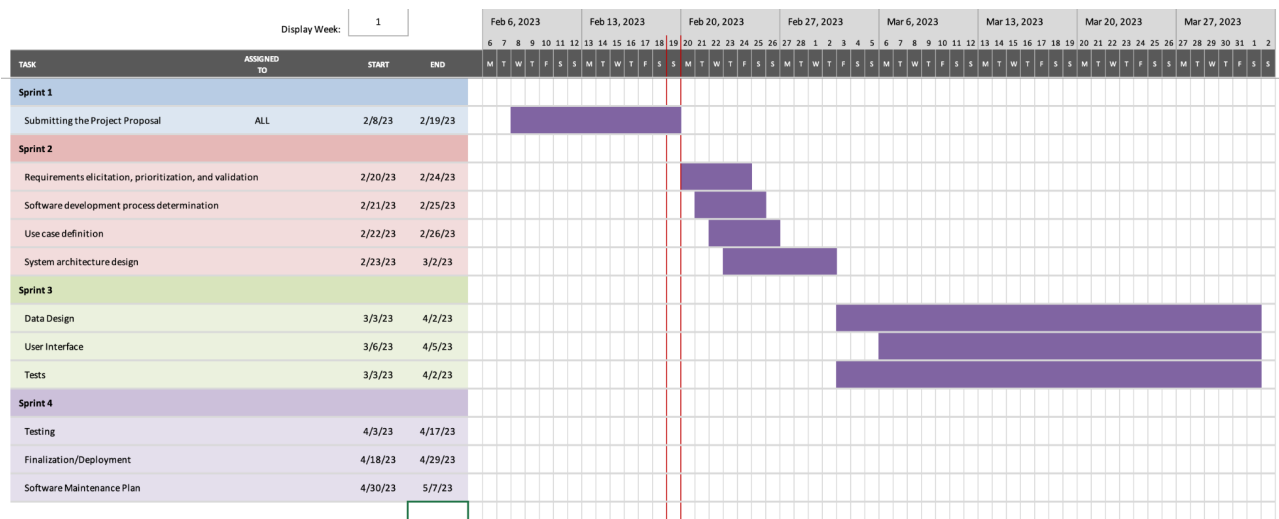
Android Studios will be used for the Java implementations of the application. A MySQL database and connector will be used to establish a connection between the mainframe and database. GitHub will be used for the team to review and commit their implementations and work, as well as an online backup of code. The team will be using the agile model to pace the progress of the application as a more optimal approach for the scope of our

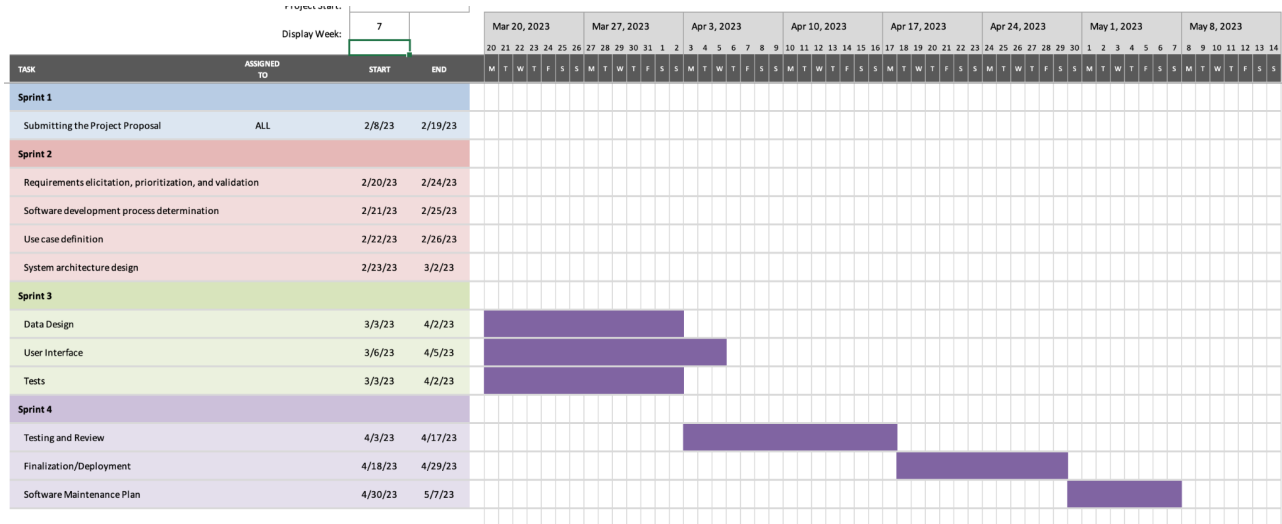
project. Discord will be used to update each team member's progress and any issues that arise. For quality assurances, we will create tests and update them as we add onto the application.

2.4 Change Management

Changes will be communicated between the team members and acquirers to be evaluated and approved if certain criteria's needs to be changed. If the change is approved, the project's schedule and quality will be resolved through scrums. Communication between the members is of utmost importance to ensure that the expected developments are done on schedule. Team members will keep each other accountable for individual shortcomings/issues.

3. Timeline

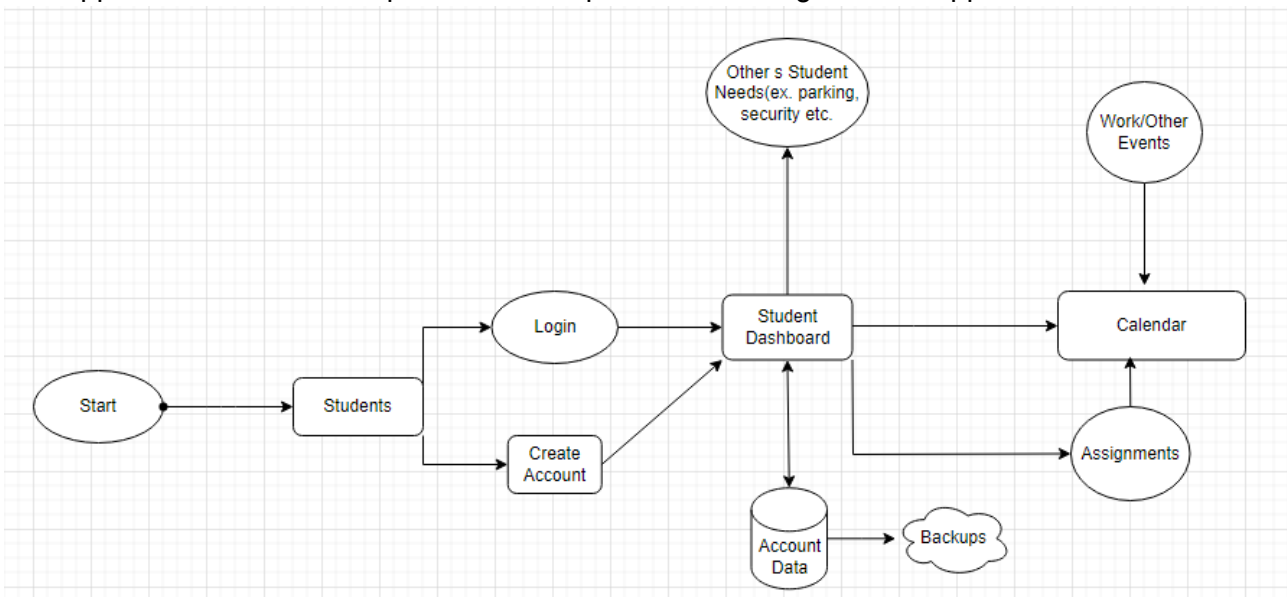




4. Overall Description

4.1 Product Perspective

This application is made to improve and/or replace the existing CSUSM app.



4.2 Product Functions

- Create school user account
- Login into account to match data
- Let's users monitor their school priorities
- Allow users to frequently update their information
- Allows users to visually see and modify their own calendar

- Displays parking slot availability within the school

4.3 Technical requirements

The product must be written in Java code. The application will require a computer/server to host the database and communicate with the application. Users will need to have a mobile device with internet access and sufficient memory space available to store the product and the data implemented.

4.4 Operating Environment

The application will operate on Android mobile devices. It will be written in Java and utilize a MySQL database and Java connector to interact with the database.

4.5 Design and Implementation Constraints

Implementation constraints in regards to the notification system as we currently do not have one, we were unable to properly implement them according to our working application.

4.6 User Documentation

The application will have a manual attached.

5. External Interface Requirements

5.1 User Interfaces

The user interface will be implemented using tools and software packages in Java in android studios. There will be UI pages where users can click on the calendar they want to view.

5.2 Hardware Interfaces

The user must have an Android supported device in order to use the application, along with a stable internet connection. The application will not have a high hardware requirement so most modern devices will be supported.

5.3 Software Interfaces

The application will utilize the Java database connector to be able to connect to a MYSQL database that will consist of the information of the user (student) and assignment data.

6. System Features

System features will include a login system, create account feature, calendar view feature, and a display of available parking features.

6.1 Login

6.1.1 Description and Priority

Access to course materials.

Priority: High

Benefit: 9

Penalty: 1

Risk: 1

Cost: 1

6.1.2 Stimulus/Response Sequences

User Action: User opens the app

System Response: Takes user to the login page where there are text fields and buttons to show where the user will type in order for them to log in

User Action: Click on Email

System Response: Let's user type in their email.

User Action: Click on password

System response: Let's user type in their password.

User Action: Click on Login

System response: The system checks the database to see if the email is in the system, and checks if the password pertaining to that email is correct or not. If correct, let's user login and , if not, access will be denied and print "unsuccessful".

6.1.3 Functional Requirements

REQ-1:The user must have an account and be able to log in using their username and password. If the information is wrong, give notice and let the user retry

REQ-2:The system must be able to check the database to see if the password and usernames are correct and match.

6.2 Create Account

6.2.1 Description and Priority

Creating a new accounts

Priority: High

Benefit: 9

Penalty: 1

Risk: 1

6.2.2 Stimulus/response Sequence

User Action: User clicks on create account button

System Response: Takes user to the create account page where there are text fields and buttons to show where the user will type in order for them to create an account

User Action: Email

System Response: Based on the input from the user, the system requires the email address to contain "@csusm.edu" to correlate the student being from csusm.

User Action: Password

System response: Based on the input from the user, the system requires the password to be at least seven characters long.

User Action: Register button

System response: When creating a user, if the email and password conditions are met, the system should add the following user's email address and password hashed into the database

6.2.3 Functional Requirements

REQ-1: Users should be able to click or have a touch screen device to click or press to input text

REQ-2: The user should be able to type into the designated text fields.

6.3 Calendar

6.3.1 Description and Priority

Calendar

Priority: High

Benefit: 9

Penalty: 1

Risk: 1

6.3.2 Stimulus/response Sequence

User Action: User navigates to calendar from Login

System Response: Displays the calendar view of events, exam dates, and assignment due dates.

User Action: User click on add event.

System Response: App shows an interface for students to add events, date, time, location and description.

User Action: User click on save.

System Response: App validates the information save the new event and adds it to the calendar, if the event has conflict with another, gives a notice.

User Action: User clicks on create event.

System Response: App shows an interface for students to add reminder name, date, time, location, and description.

User Action: User clicks on save.

System Response: App validates the information and saves the new data onto the server to the user when the saved date and time arrives with a description.

6.3.3 Functional Requirements

REQ-1: User to click on the desired date.

REQ-2: User to be able to type into designated textfield in order to input the data into the calendar to store into the database.

6.4 Parking Availability Display

6.4.1 Description and Priority

Shows the user the different parking lots along with how many parking spots are available.

Priority: High

Benefit: 9

Penalty: 1

Risk: 1

6.4.2 Stimulus/response Sequence

User Action: User navigates to parking from the homepage

System Response: Displays the parking lot list of the parking lot names with the availability next to them.

User Action: Go back button

System Response: User returns back to the homepage

6.4.3 Functional Requirements

REQ-1: Users should be able to click or have a touch screen device to click the parking button.

7. Other Nonfunctional Requirements

7.1 Performance Requirements

The software should respond to user inputs, such as button clicks and text field inputs. The app should be able to save and update changes made by the users. The software should be available and usable to users 24/7, except while under scheduled maintenance. The software should be run smoothly across mobile devices and any browsers.

7.2 Safety Requirements

The software will follow the CCPA(California Consumer Privacy Act) to protect users personal information and ensure that the data collected and stored are protected securely. The data recorded will not be sent to any third party services.

7.3 Security Requirements

The software will have a secure hash encryption algorithm transmitted between the user and the server. Without proper username and password specifications, the user won't be able to enter his/her profile.

7.4 Software Quality Attributes

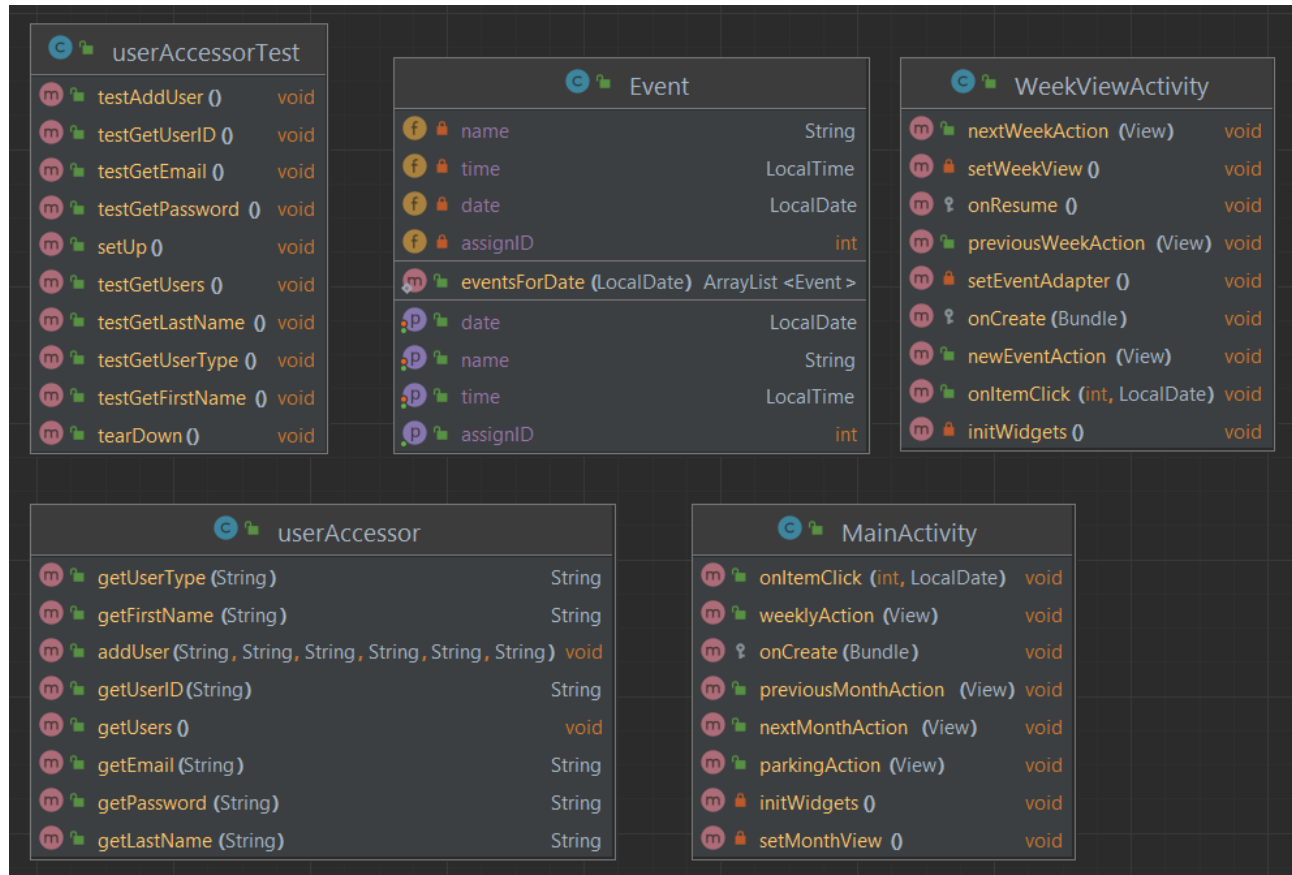
- 7.4.1 Adaptability: Users will be able edit their inputted data
- 7.4.2 Availability: The application will be available to users 24/7.
- 7.4.3 Correctness: The application should meet its functional requirements.
- 7.4.4 Flexibility: Changes and modifications will be tested and run over an android emulator in order to provide any improvements necessary.
- 7.4.5 Interoperability: The software should be compatible with different platforms and devices.
- 7.4.6 Maintainability: The application should be easy to update and troubleshoot.
- 7.4.7 Portability: The application will be compatible with different devices and operating systems.
- 7.4.8 Reliability: The application should not crash under normal operation and repeated inputs.
- 7.4.9 Reusability: Parts of the codebase should be reusable for future add-ons.
- 7.4.10 Robustness: The software should be able to handle unexpected inputs and errors without causing a crash or data loss in the system.
- 7.4.11 Testability: Testings will be run using automated testers over an emulation.
- 7.4.12 Usability: The software will be user friendly and easy to navigate.

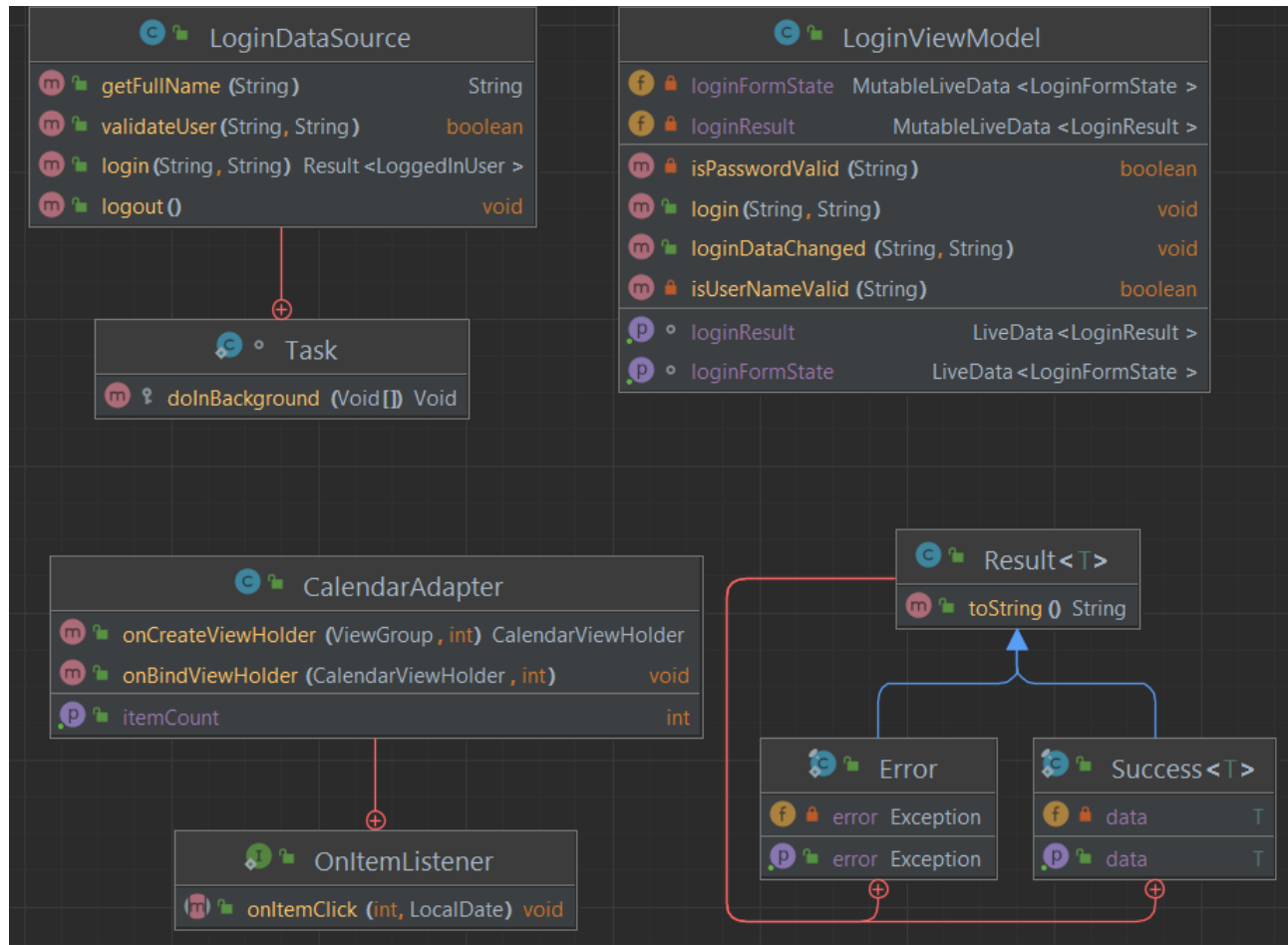
7.5 Business Rules

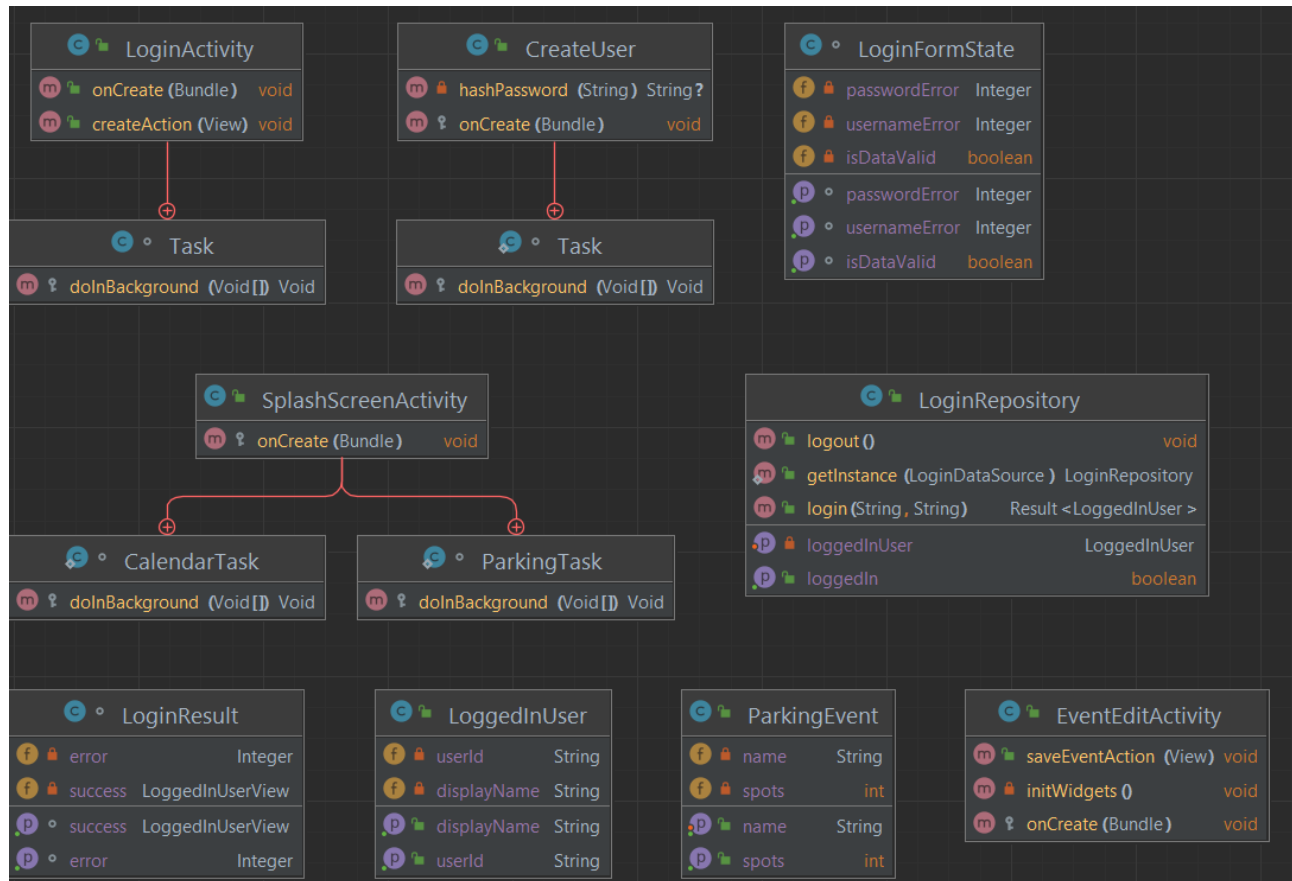
Only the user of their personal account may edit, change or modify their data accordingly. With the protection and security of the program to the databases, the user may only change what's theirs.

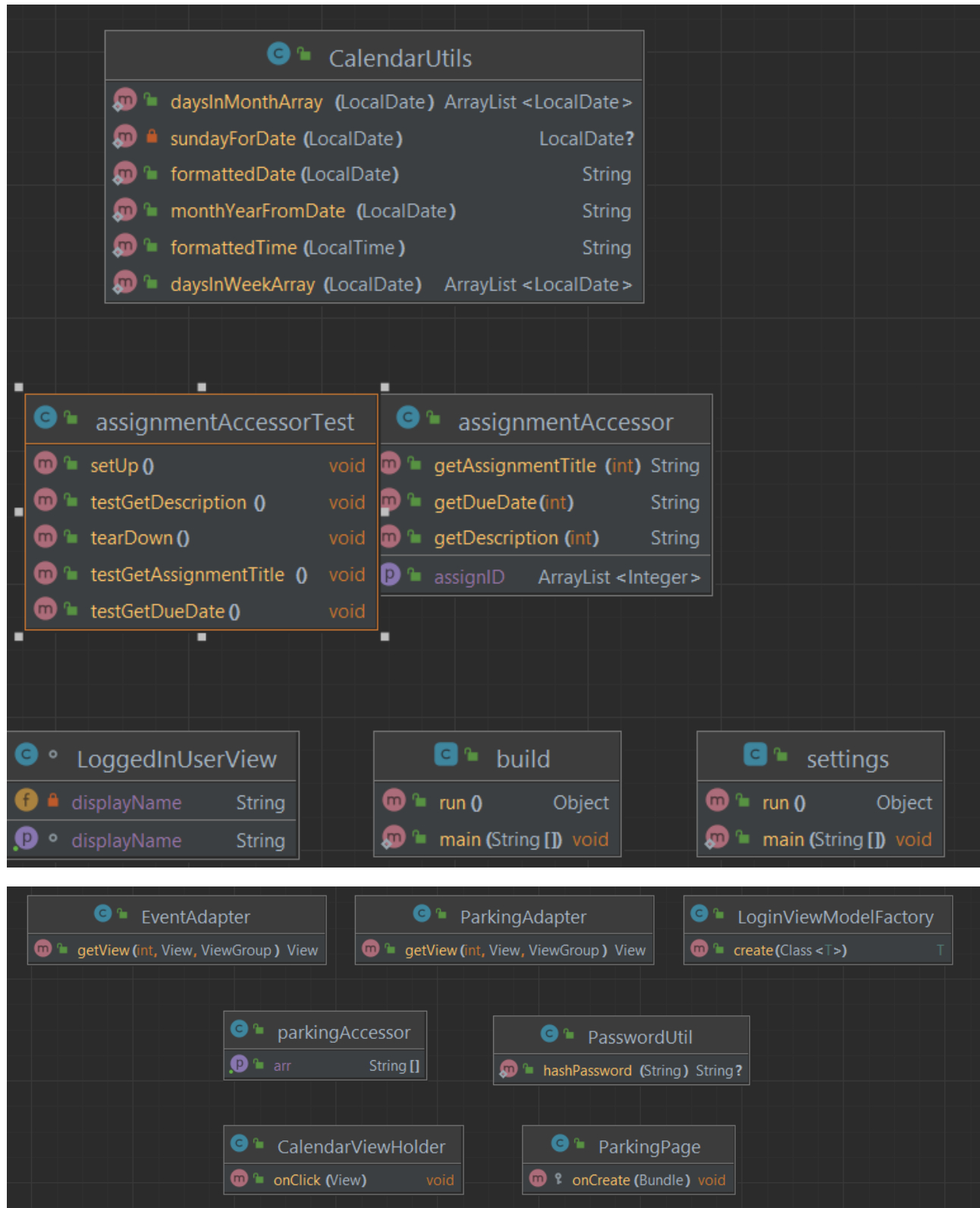
8. System Architecture

8.1 Architectural Design





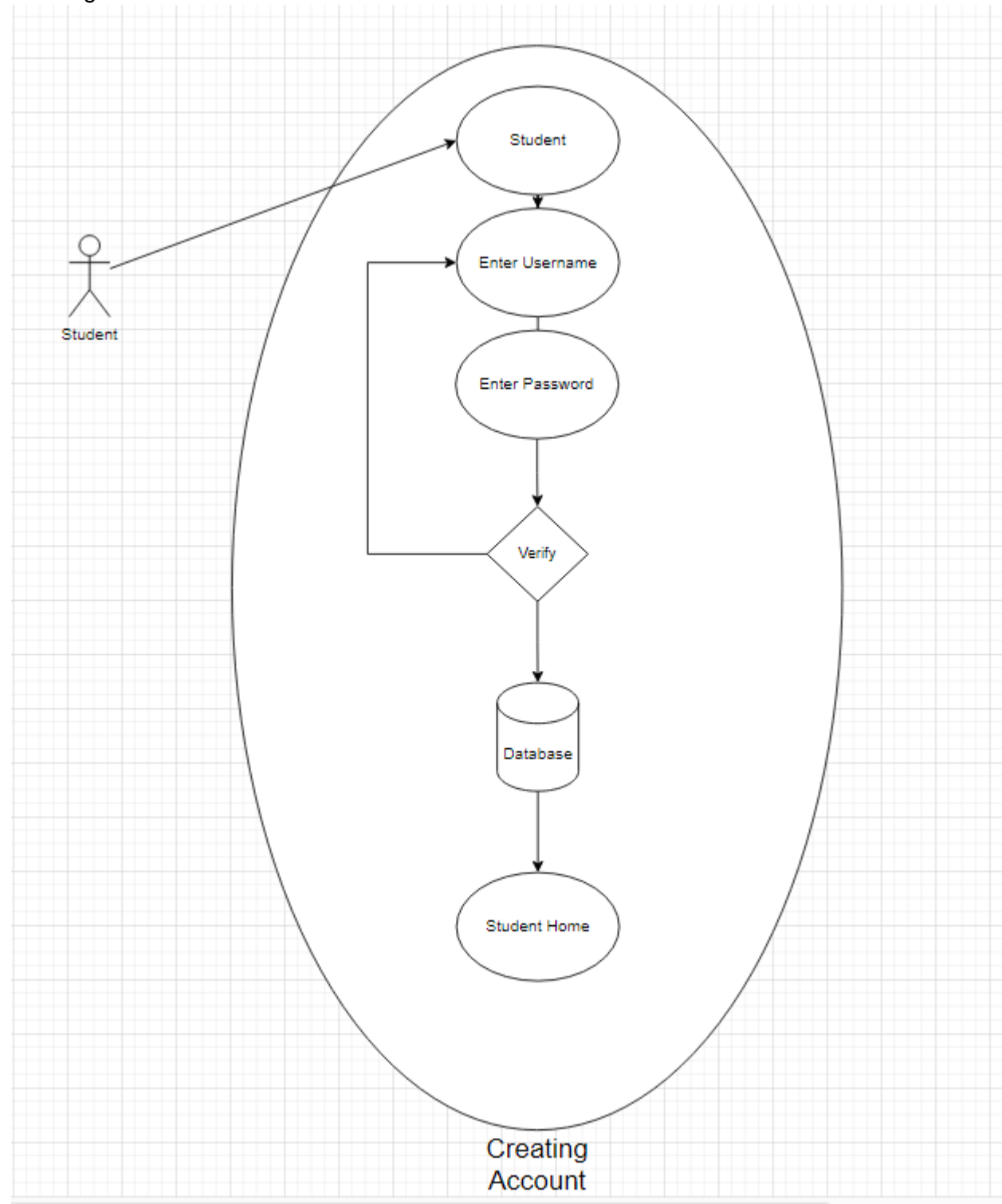




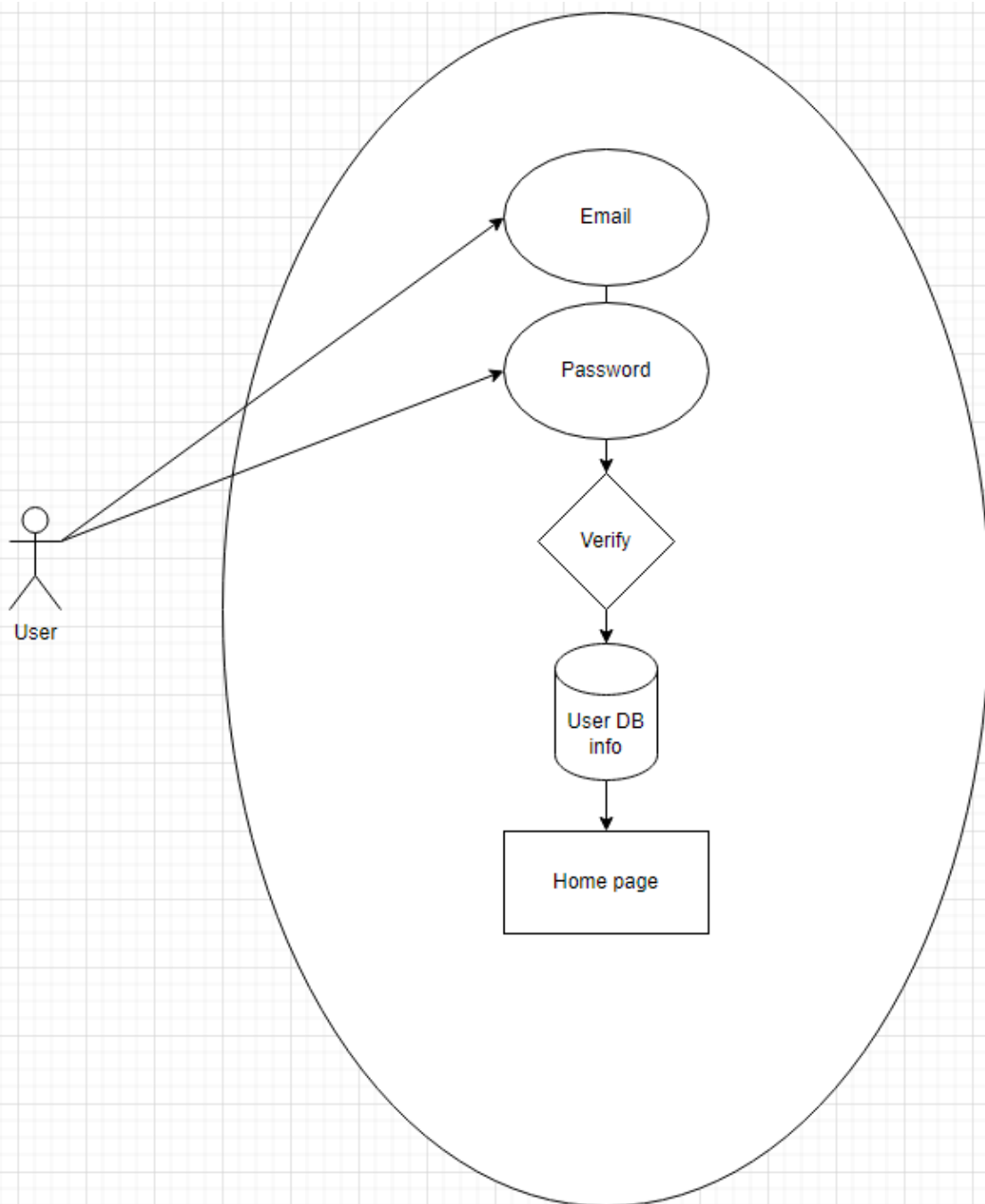
All modules will draw and update information to and from the database, allowing for each individual piece to always be constantly updated and work together

8.2 Decomposition Design

Creating Student Account

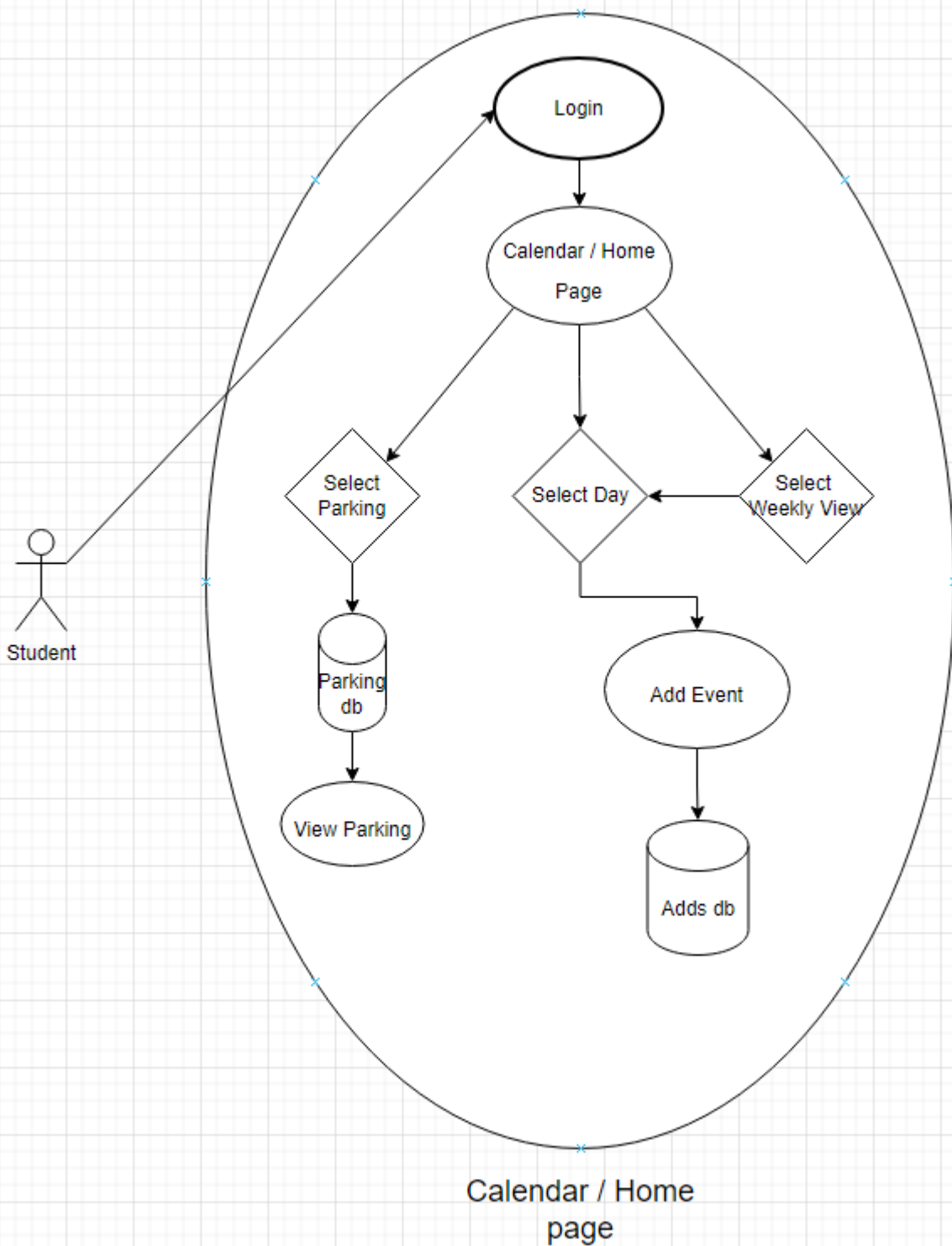


Login

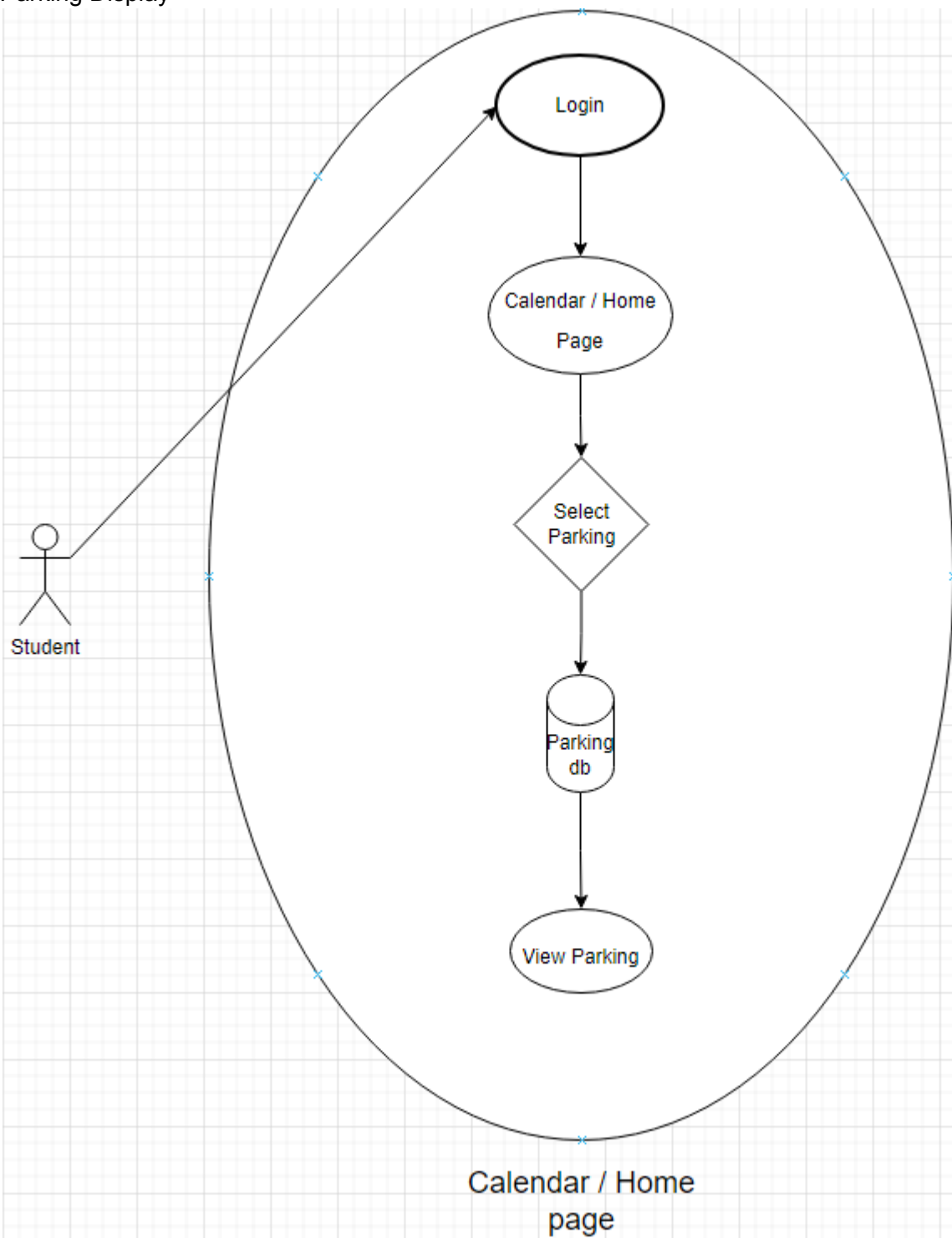


Login

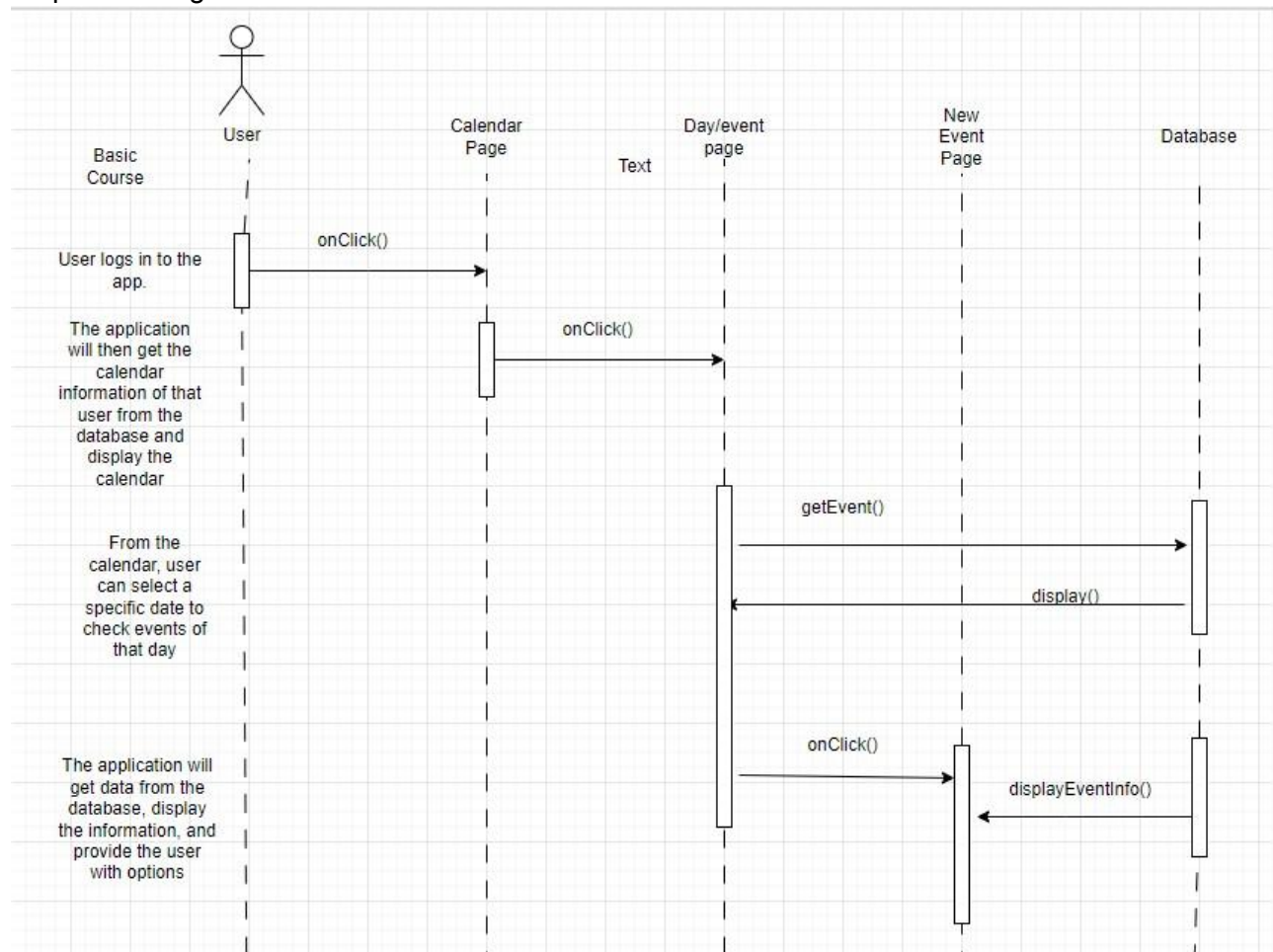
Calendar / Home Page



Parking Display



Sequence Diagram

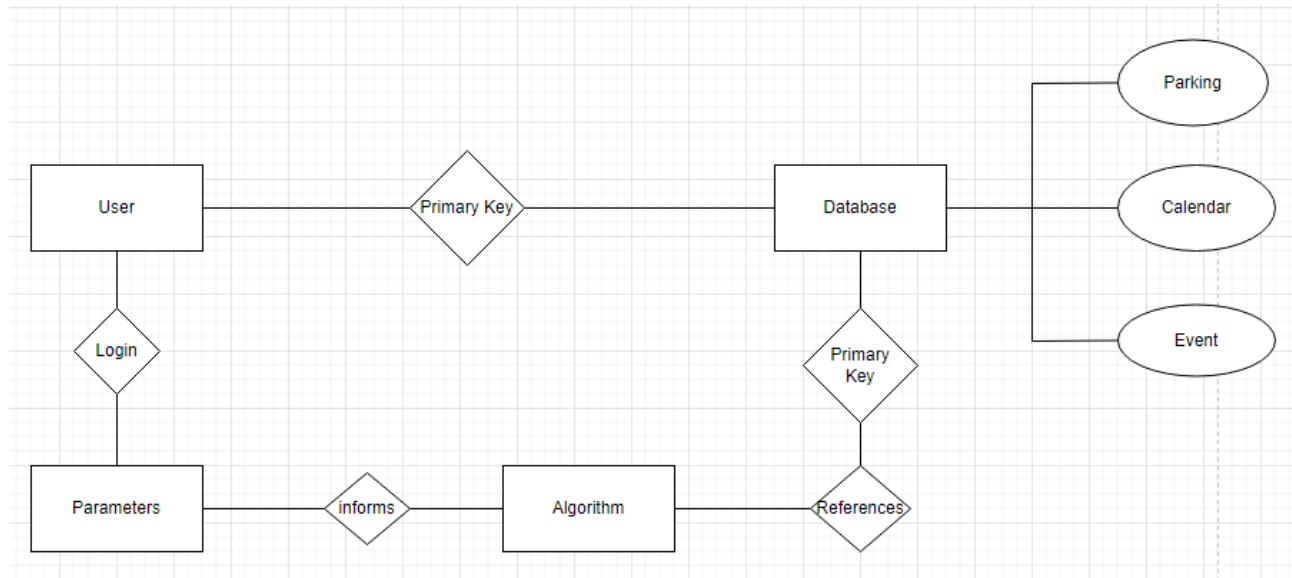


8.3 Design Rationale

We selected that architectural design because it was best fit for our vision of the application. With how our application is being implemented, the uml diagram was how our plan can be accurately visualized. Issues we may encounter are the dependencies between each separate feature may interfere with one another. However we trade with the accuracy and accessibility of each feature and the breakdown in order to create less errors

9. Data Design (only applicable if your project has a database component)

9.1 Entity relationship design



9.2 Data dictionary

Entity	Attributes	Description
User	userID, firstName, email, role()	store login information of users
Login	userID, password	information required to log in
Inform	messageType	return message for entry
Reference	title, URL	SQL reference material
Database	databaseID, Name, Server	database used by the application
Dashboard	edit, class, assignment	user-specific design and data
Courses	Title, Instructor, startDate, endDate	course information for users
Assignments	assignmentID, courseID, title, dueDate	store class assignments
Calender	startDate, endDate, time()	calender event for users

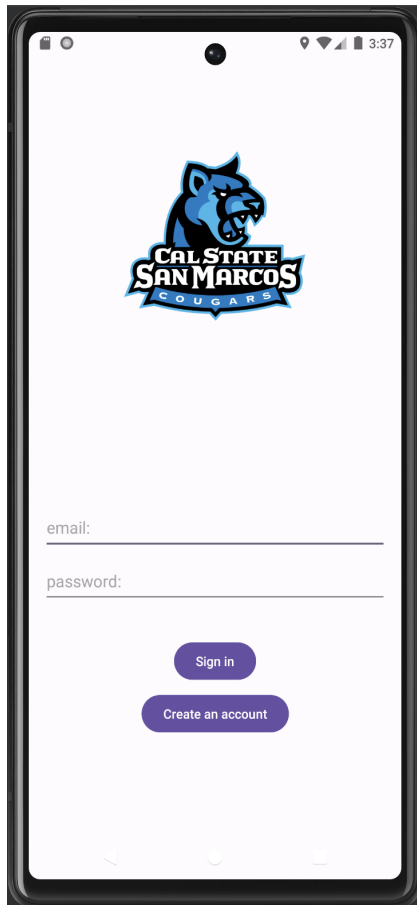
10. Other Requirements

Appendix A: Screen Images

1.



2.



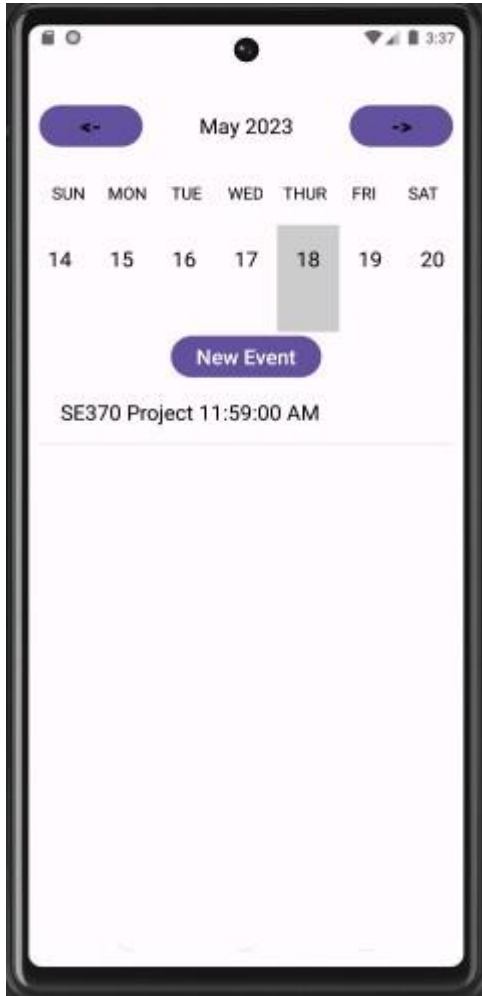
3.



4.



5.



6.



7.



Image Descriptions:

Image 1 - The splash screen image of csusm's university logo.

Image 2 - The login page. Within this page consists of csusm's mascot logo splash image, a text field labeled "email", another text field below that labeled "password", a button labeled "Sign In", and a button labeled "Create an account". The user, if already made an account, will have to input their email and password into the designated text fields by clicking the "Sign in" button in order to access their account and their data accordingly with the app. However, if the user does not have an account, the user will then have to click the "Create an account" button in which the user will be directed to the Create Account page (image 3).

Image 3 - Create an account page. When at the create an account page, the user will be met with the csusm's mascot logo splash image, a text field labeled "email", another text field below that labeled "password", and a button labeled "Register". The user will have to register by inputting a valid csusm email and a valid password at least seven(7) characters long. If and when the criteria for the email and password is met, the user then will click the "Register" button then the system will save the email with the password, hash and encrypt the password saving the user into the database so the user may access their account when logged in saving any information they input.

Image 4 - Calendar / Home page. This page consists of a calendar view, a button on the top center labeled "Weekly", two(2) arrow buttons on the top left and top right of the calendar view, and a button at the bottom center labeled "Parking Info". The "weekly" button directs the user to the weekly view page(image 5). The left and right arrow buttons when clicked allow the user to view different months. The "parking info" button when clicked directs the user to the parking information page (image 7). Within the main calendar, the user may select the date to which they desire to add any assignment or event information accordingly(image 6).

Image 5 - Weekly page. This weekly consists of two(2) arrow buttons, left and right, that allows the user to view different weeks of the month. Below that is the calendar view of just the desired week the user wishes to view. And below the weekly view is a "New Event" button which when clicked allows the user to input information about a new event they want to add to that date they select(image 6). The information will be displayed in view fields always below the "New Event" button allowing the user to create multiple events on one day.

Image 6 - New Event page. The new event page consists of the "name" textfield, "name" title text, the date of which the user selected to add a new event on, the time, and a "save" button below and centered. When the user is directed to this page, the user will then be able to input text into the text field to store in the new event data they wish to store on that date. After the user has typed the desired information, they may click the save button in order to save the inputted text into the designated date. When the user clicks "Save" the user's info will be saved into their data within the database so that the specific user may only see and the information then is displayed onto the calendar back on the calendar/home page(image 4).

Image 7 - Parking lot availability page. The parking lot availability page is a very straightforward page as it consists of the title of the page, "Parking Lot Availability", and the information of each parking lot as a list. From left to right it shows the lot name, number of available parking, and how many parking spaces total there are in the specific parking lot. This list shows csusm's parking lot information from lots B, C, F, K, XYZ, and PS1.

11. Github Link

<https://github.com/williamphong/CSUSMStudentApp>

12. Evolution

12.1 Software

We changed from the use of the IDE intellij to android studios in order to fulfill the mobile application portion and be able to test our program in real time with our code. This allowed us to have a better understanding and usage of UI.

12.2 Time and Resource

Importation of different Java frameworks allowed us to have a start of an understanding how UI communicates with the code in order to achieve certain tasks. The lectures as well as multiple open source tutorials allowed us to learn and apply what we can for our application.

12.3 Time Distribution

Most of our time was spent on figuring out what features we wanted to make, the flow of how we wanted to make them, as well as how to incorporate each feature so that it all flows together accordingly. After months of preparation, the final month and a half from April to May was spent coding each feature then combining everything into the final product.

12.4 Software Change Management

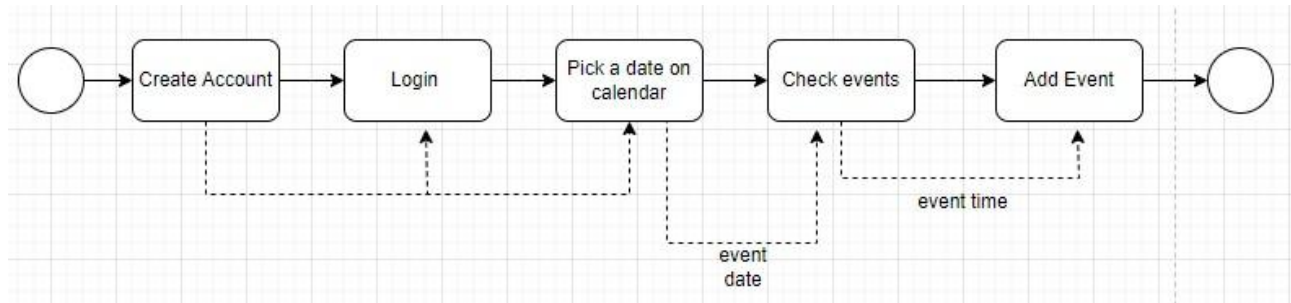
The change from using intellij as our main coding IDE to android studios significantly changed our timeline as we were learning an unfamiliar IDE and applying it to our project. The use of gradle instead of maven was favored as it helped progress the project towards our goals for each feature connecting with one another.

12.5 Learnings and Limitations

From this project, we learned what it takes to start becoming a software engineer, the mindset of a software engineer, and the challenges that come with it. From the ethics, project and team management, to the different software lectures, we learned how to properly start and finish a starter software application. Some limits we have had throughout the project were figuring out new IDEs as we went along and how to properly sync and connect each different feature to one another so that they all work properly to flow well.

13. Workflow Diagram

Workflow Diagram for adding an event



14. User Manual

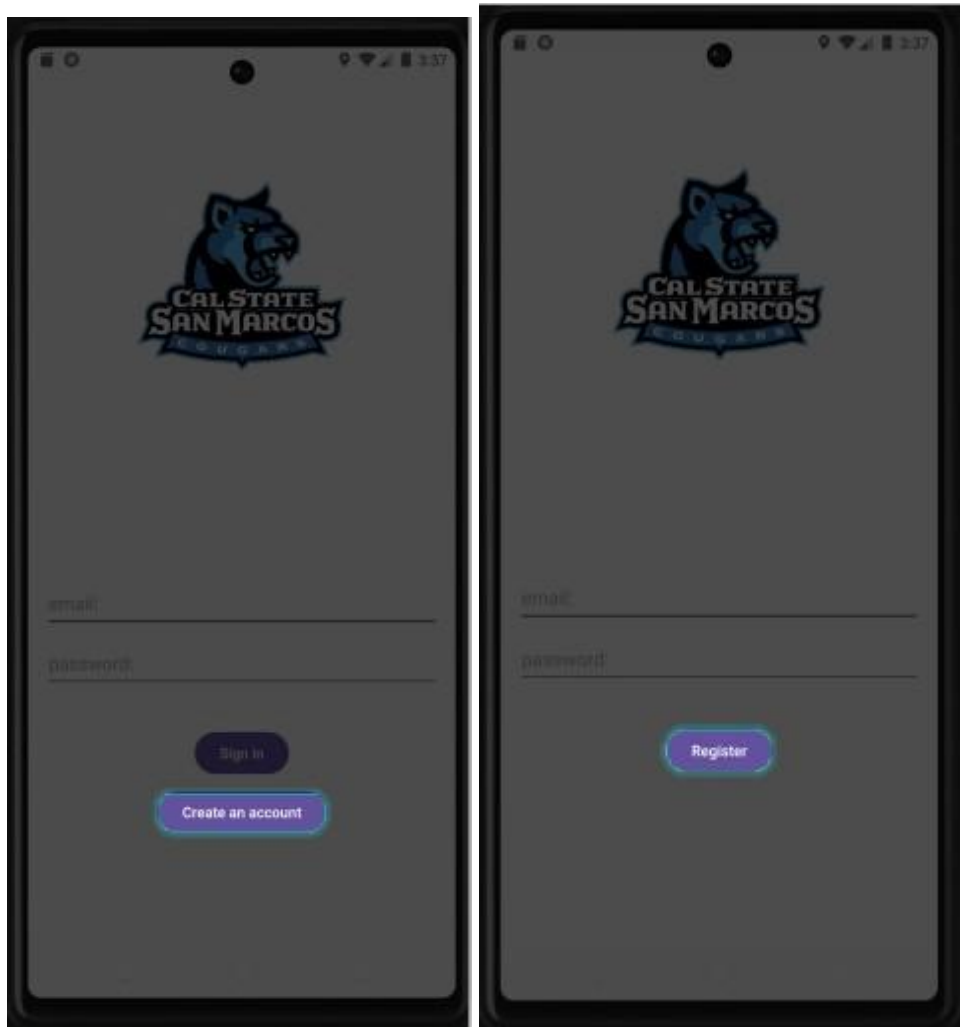
14.1 Getting Started

Requirements

To use this app, students must have an android supported device and connection to the database.

Account Creation

Users can create an account by clicking on the “Create an Account” button when the user opens the app on the Login Page.



Once the user clicks on “create account” the create account page will be shown and let’s the user enter an email (the email needs to be an @csusm email) and a password at least 7 characters long. After inputting the required information, the user will be able to register their account by clicking on the register button.

Logging In

From the main screen, the user can input their registered email and password and click “Sign In” to sign into their account



14.2 Features

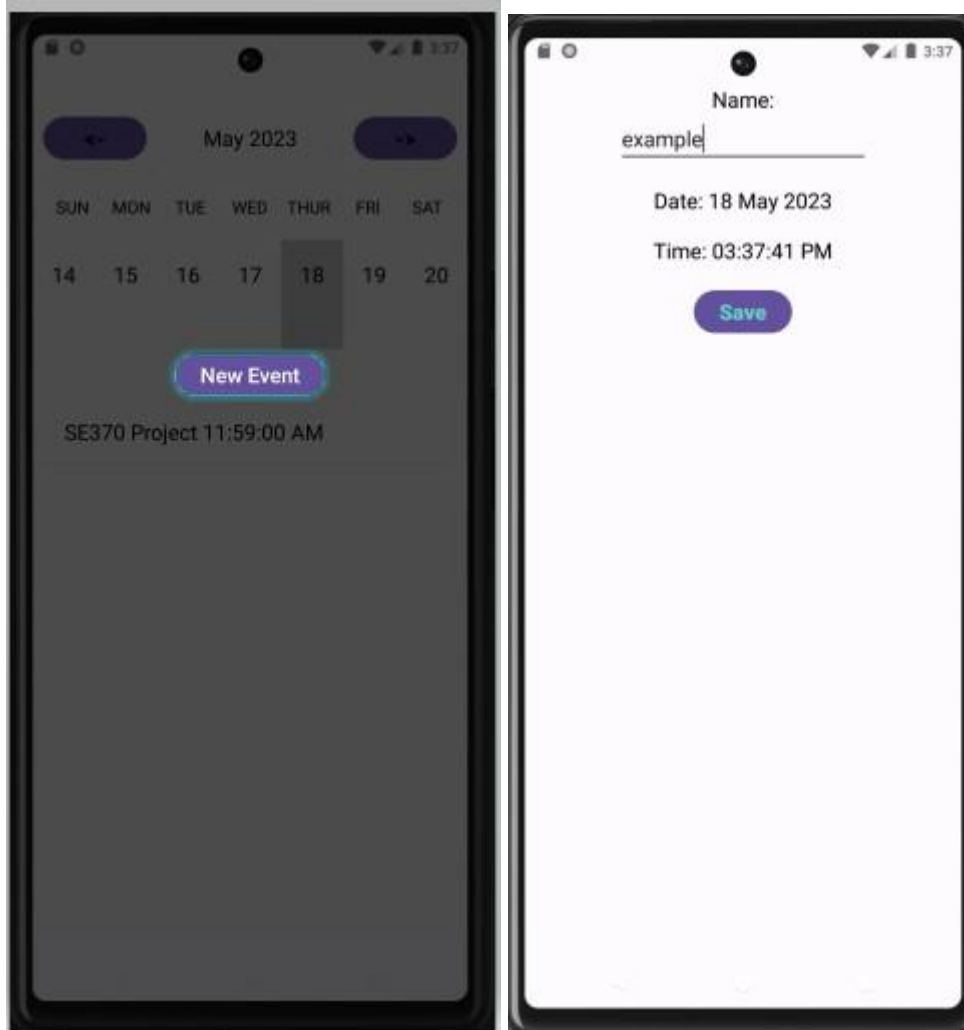
Calendar

The calendar shows a calendar on a monthly view. If the user wants to change it to a weekly view, they can click on the “Weekly” button at the top of the screen.



Events

The user can see the events by clicking on a date in the calendar and it will show the information of that event. If the user wants to add an event, they can also do that by clicking on the “New Event”, from there users can input the name of the event and save it.



Parking

The user can check the parking lot information by clicking on the “Parking Lots” button at the bottom of the calendar.



Exiting

If users want to close the app on an android mobile device, they will need to manually close the app from the android.