Scheduling Flutter App

By Hannah Jeffers, Matthew Latta, Patrick McManus, Michael Connors, and Pedro Gomez

# Table of Contents

[Table of Contents 2](#_Toc1923107285)

[1). Introduction 2](#_Toc913509299)

[1.1 Purpose of the Document 2](#_Toc1913130596)

[1.2 Intended Audience 2](#_Toc145803227)

[2). General Description 3](#_Toc343278139)

[2.1 Product Goal 3](#_Toc1496998223)

[2.2 Product Profile 4](#_Toc1102147459)

[2.3 Assumptions and Dependencies 4](#_Toc361275598)

[3). Functional Requirements 5](#_Toc2111269077)

[3.1 Profile Creation 5](#_Toc1649560335)

[3.1.1 Log In 6](#_Toc450747189)

[3.1.2 Forget and Reset Password 6](#_Toc648464976)

[3.1.3 Sign Up 7](#_Toc1152177741)

[3.1.5 Profile Database 7](#_Toc2081296118)

[3.2 Coordinator Page 8](#_Toc514413626)

[3.2.1 Create Event 8](#_Toc24337241)

[3.2.2 Delete Event 8](#_Toc1678467372)

[3.2.3 Schedule and Post Event 9](#_Toc1405473939)

[3.3 Calendar 9](#_Toc1782970611)

[3.3.1 Create calendar 9](#_Toc130375574)

[3.3.2 Delete calendar 9](#_Toc1490866871)

[3.3.3 Add event to calendar 10](#_Toc1297616069)

[3.3.4 Upload Calendar 10](#_Toc1931615329)

[3.3.5 Share Calendar 11](#_Toc1825964752)

[3.3.6 Combine Calendar 11](#_Toc263482380)

[4). Non-Functional Requirements 12](#_Toc280406918)

[4.1 Requirements on environment 12](#_Toc1206718870)

[4.2 Requirements on error conditions 12](#_Toc1280510150)

[4.3 Gantt Chart 12](#_Toc745379352)

[5). Feature Evolution 13](#_Toc498027324)

[Appendix A – Definitions 15](#_Toc1286737115)

[Appendix B – Abbreviations 15](#_Toc1085944521)

[Appendix C – References 15](#_Toc1456096180)

# 1). Introduction

## 1.1 Purpose of the Document

This document identifies technical requirements related to a scheduling and event coordinating application.

## 1.2 Intended Audience

The intended audience of this document is primarily college students, faculty, & staff.

Testers will also utilize this document when creating test case plan.

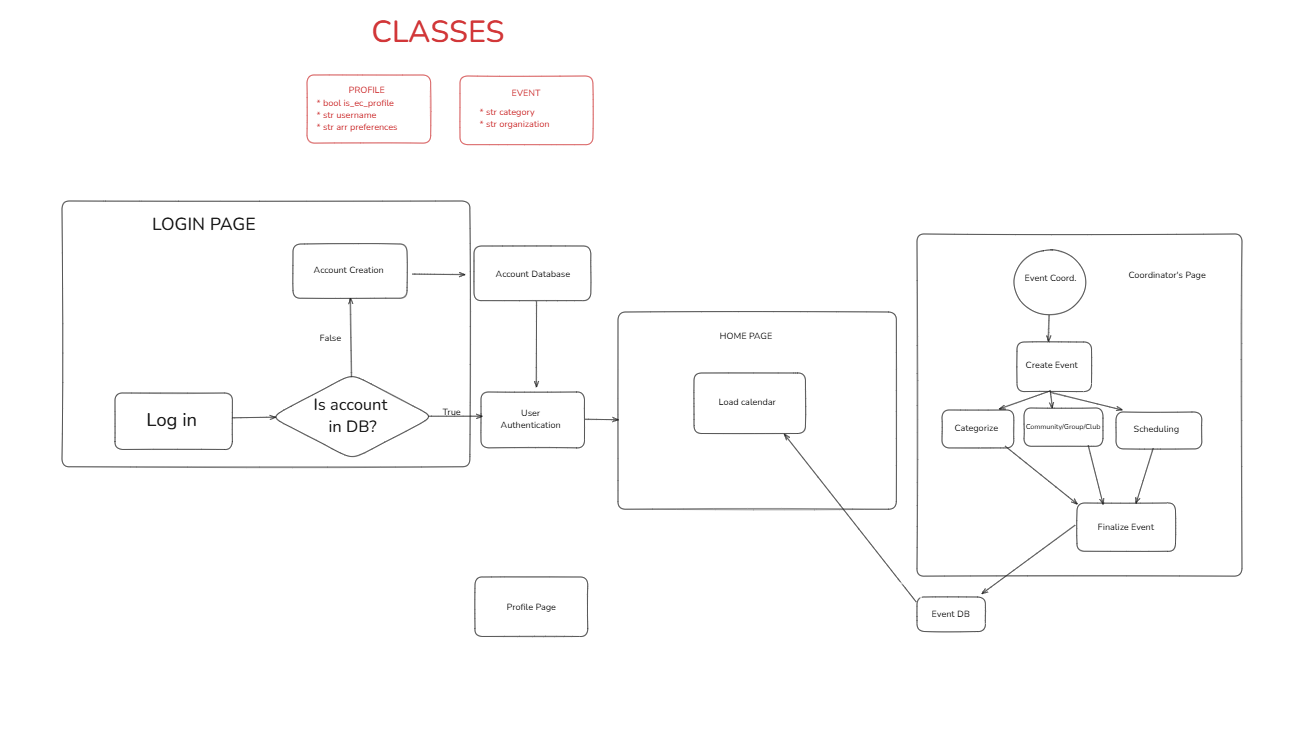
# 2). General Description

## 2.1 Product Goal

The scheduling app aims to resolve issues revolving around organization/club searches, event planning and student engagement outreach. It will spread information about events just like the current system in Outlook, except it will be in a place that makes it much easier to manage schedules and does not keep you from your important emails. It is similar to the project of another group in CS-375, but it focuses more on easing the burden of event coordinators rather than to make scheduling easier for students. To our scheduling app, the help for students is more of a side effect of easily managed schedules for event hosts.

## 2.2 Product Profile

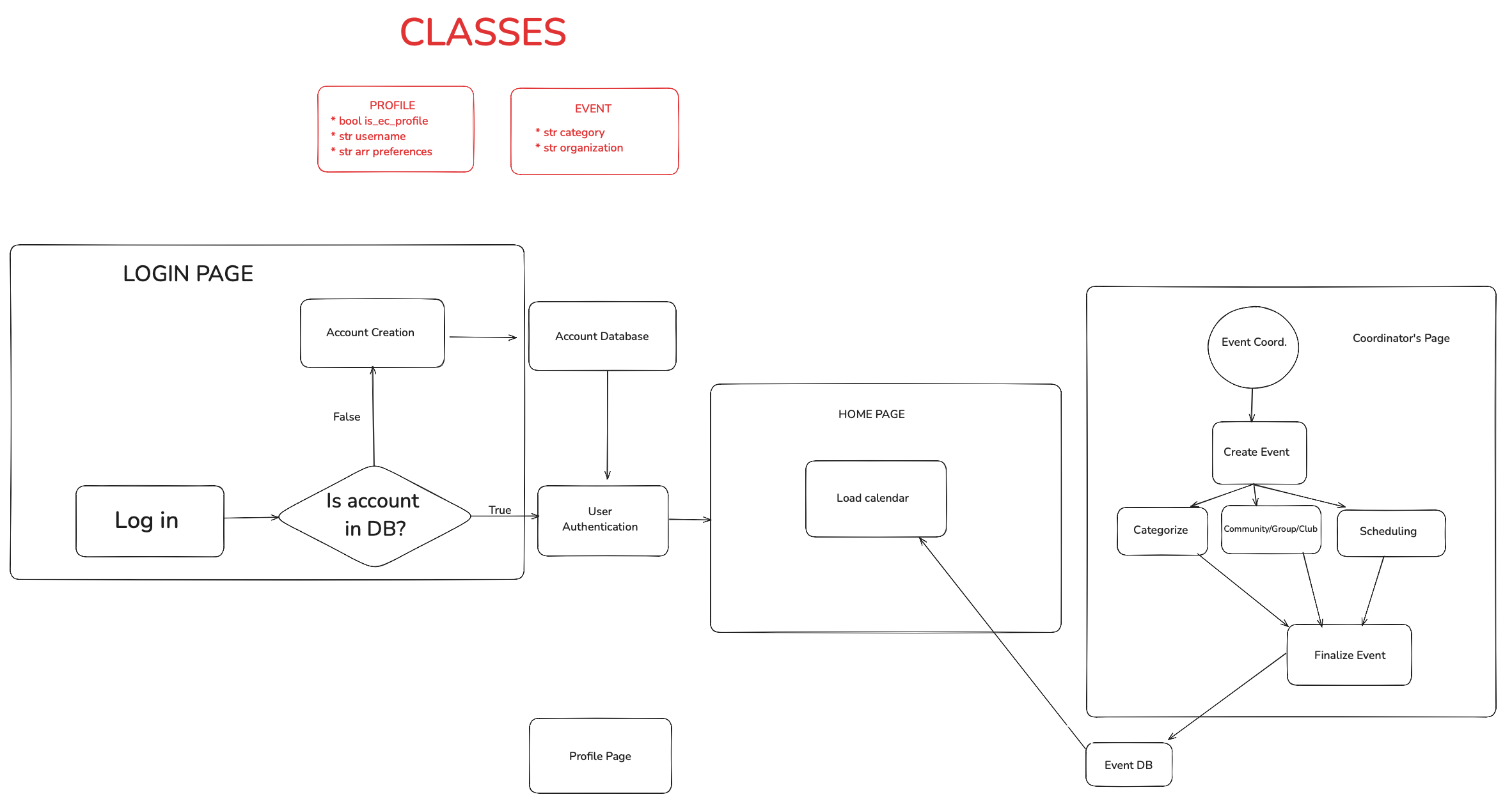
The following diagram gives an overview of the functionality of the scheduling app. On the login in page, it must verify if the user has an account already stored in the database. If not, then it guides the user through the account creation process and stores the new account for future use. Afterwards, the user must be authenticated before entering the home page. Once accessed, the home page can load the calendar of events and allow the user to partake in various functions to customize their calendar to their pleasing. The events added to their calendar come from a database of events. The event coordinator side of the app allows club leaders to create their own events. They will need to categorize the event and add a schedule for this event to be finalized and added to the event database that the user can use for their calendar customization.



## 2.3 Assumptions and Dependencies

Our scheduling app depends on the user having an iOS or Android mobile device that is capable of running the app and has it installed. This includes enough memory and processing power to run the app on your mobile device.

# 3). Functional Requirements



## 3.1 Profile Creation

Profile Creation

The app allows new and old users to create a profile to access the app.

### 3.1.1 Log In

login

The app has the ability to allow a profile to login to the app.

login\_command

The app has a login command.

login\_command\_dialog

The app displays a dialog box containing two input text boxes for email and password.

login\_command\_submit

The app has a login submit.

### 3.1.2 Forget and Reset Password

forget\_reset\_password

The app has the ability to allow a profile to reset their password if they forget it.

fr\_password\_command

The app has a forget and reset password command.

fr\_password\_command\_dialog

The app displays a dialog box containing one input text box for the email that the profile is connected to.

fr\_password\_email

The app sends an email to the input email with a link to reset the password.

fr\_password\_link

A hyperlink that directs to the app’s reset password page.

fr\_password\_link\_dialog

The app displays a dialog box containing one input text box for a new password.

### 3.1.3 Sign Up

signup

The app can allow a new profile to be created for the app.

signup\_command

The app has a signup command.

signup\_command\_dialog

The app displays a dialog box containing four input text boxes for email, password, first name, and last name, and one dropdown menu for user type.

signup\_command\_submit

The app has a signup submit.

3.1.4 Create Profile

Profile

The Profile class allows for the creation of new user profiles.

profile\_create\_pf

The class creates and returns a new profile object.

profile\_add\_preference

The class allows profiles to add preferences to filter their event/club suggestions.

profile\_remove\_preference

The class allows profiles to remove preference from their preferences list.

### 3.1.5 Profile Database

database

The app can store profile information in a class-object database.

db\_store\_profile

The app can store new profiles into a database.

db\_remove\_profile

The app can remove profiles from the database.

db\_authenticate\_user

The app uses the database for user authentication.

## 3.2 Coordinator Page

Coordinator Page

Coordinator page where the coordinator tools exist.

### 3.2.1 Create Event

cp\_create\_event

The app can create an event.

cp\_create\_event\_command

The app has a create event command.

cp\_create\_event\_dialog

The app displays a dialog box containing three input text boxes for event title, event description, and event category.

cp\_create\_event\_submit

The app has a create event submit.

### 3.2.2 Delete Event

cp\_delete\_event

The app can delete an event.

cp\_delete\_event\_command

The app has a delete event command.

cp\_delete\_event\_dialog

The app displays a dialog box containing a yes/no option to confirm if the user wants to delete the event.

cp\_delete\_event\_submit

The app has a delete event submit.

### 3.2.3 Schedule and Post Event

cp\_schedule\_event

The app can schedule an event.

cp\_schedule\_event\_command

The app has a schedule event command.

cp\_schedule\_event\_dialog

The app displays a dialog box containing one dropdown menu for created events with no schedule and two input text boxes for date and time.

cp\_post\_event\_submit

The app has a post event submit.

## 3.3 Calendar

Calendar

Calendar page where the calendar tools exist.

### 3.3.1 Create calendar

c\_create\_calendar

The app can create a calendar.

c\_create\_calendar\_command

The app has a create calendar command.

c\_create\_calendar\_view

The app displays a view of a blank calendar.

c\_create\_calendar\_dialog

The app displays a dialog box containing one input text box for calendar title.

c\_create\_calendar\_submit

The app has a create calendar submit.

### 3.3.2 Delete calendar

c\_delete\_calendar

The app can delete a calendar.

c\_delete\_calendar\_command

The app has a delete event command.

c\_delete\_calendar\_dialog

The app displays a dialog box containing a yes/no option to confirm if the user wants to delete the calendar.

c\_delete\_calendar\_submit

The app has a delete event submit.

### 3.3.3 Add event to calendar

c\_add\_event\_calendar

The app can add an event to calendar.

c\_add\_event\_calendar\_command

The app has an add event to calendar command.

c\_add\_event\_calendar\_dialog

The app displays a dialog box containing one dropdown menu for created and scheduled events that have not been already added to the calendar.

c\_add\_event\_calendar\_submit

The app has an add event to calendar submit.

### 3.3.4 Upload Calendar

c\_upload\_calendar

The app can upload a pre-existing calendar.

c\_upload\_calendar\_command

The app has an upload calendar command.

c\_upload\_calendar\_dialog

The app displays a dialog box containing a file browser window that match saved calendars file types in response to the upload command.

c\_upload\_calendar\_submit

The app has an upload calendar submit.

### 3.3.5 Share Calendar

c\_share\_calendar

The app can share a calendar.

c\_share\_calendar\_command

The app has a share calendar command.

c\_share\_calendar\_dialog

The app displays a dialog box containing options of where to save and/or send the calendar to.

c\_share\_calendar\_submit

The app has a share calendar submit.

### 3.3.6 Combine Calendar

c\_combine\_calendar

The app can combine two calendars.

c\_combine\_calendar\_command

The app has a combine calendar command.

c\_combine\_calendar\_dialog

The app displays a dialog box containing two dropdown menus of saved calendars to combine.

c\_combine\_calendar\_submit

The app has a combine calendar submit.

# 4). Non-Functional Requirements

## 4.1 Requirements on environment

app\_platform

The app runs on iOS and Android.

app\_memory

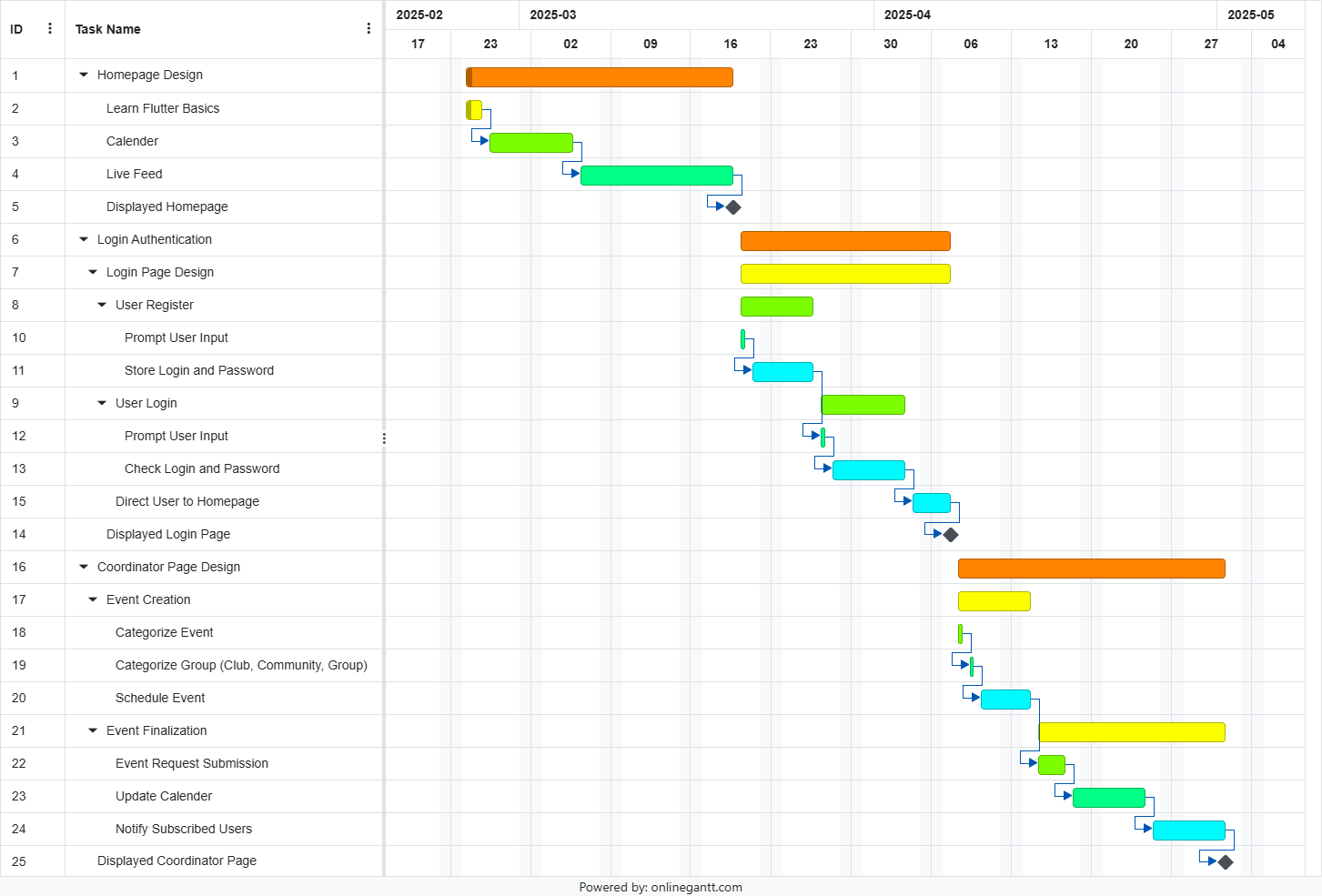
The app requires some free memory storage.

## 4.2 Requirements on error conditions

app\_stability

The app is not directly responsible for an operating system crash.

## 4.3 Gantt Chart



# 5). Feature Evolution

It is understood that all the functionality outlined by the requirements of this document will continue to exist in future versions of this application. New functional requirements may be introduced as determined by the need of users and developers who may be continuing the development of this application.

# Appendix A – Definitions

Gantt Chart – A chart used to show the distribution of effort along a scheduled time period.

Home Page – The reoccurring screen on the app that acts as a hub for navigating the most general functions.

Subscribe – When a user shows interest in an event, they can choose to receive automatic updates from its coordinator.

Authentication – When a user attempts to sign into the app, their account and password must be verified to be correct and in the database before they are allowed usage.

# Appendix B – Abbreviations

fr – forget\_reset

db – database

cp – coordinator\_profile

c - calendar

# Appendix C – References

Bill Mongan *NLiVE Requirement Specification v. 1.0* Microsoft Press, 2005.