Bilkent University

Department of Computer Engineering

CS319 Term Project

The program: Student Club Manager

Project Design Report

İdil YILMAZ - 21703556
Doğukan Ertunga KURNAZ - 21702331
Turgut Alp EDIS - 21702587
Taha Batur ŞENLI- 21901857
Argun MURADOV - 21901305
Cemhan Kaan ÖZALTAN - 21902695

Instructor: Eray Tüzün
Teaching Assistant(s): Elgun Jabrayilzade, Emre Sülün, Erdem Tuna, Muhammad Umair Ahmed and Cevat Aykan Sevinç

Nov 26, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Object-Oriented Software Engineering course CS319.

Contents

1 Introduction

1.1 Purpose of the System

The program is a web-based student club system which aims to help manage student clubs and handle various interactions between students and clubs. It allows student clubs to easily organize and promote events while also ensuring a clear communication between the Student Activities Center (SAC) and student clubs. The users in the program are the students, student clubs and the SAC Admin.

In terms of features, the program allows students to see all the activities organized by clubs, receive notifications about specific activities and customize their profile. Additionally, students can join or leave student clubs and activities. Student clubs, on the other hand, can organize or delete activities, manage their expenses, notify their members, customize their profile and manage their budget which is assigned based on how many members they have. Finally, the SAC Admin can add or delete clubs and students, change club budgets and approve or refuse events.

1.2 Design Goals

The design goals were determined using the requirement report’s non-functional requirements. The program needs to be easy to use, because it is designed to be a straightforward alternative to club management. Additionally, due to the fluctuating nature of Bilkent University student clubs, the system needs to be easily maintained by academic personnel.

1.2.1 Usability

The system will be potentially used by all Bilkent University students, so the software needs to provide ease to use. Besides, the user can learn the usage of the system easily. The usage of the system is kept on it's the simplest level. By not adding very complex functions and operations for the users who possibly do not understand the websites much to the software, it is possible to provide users a better experience.

1.2.2 Maintainability

When the users of the program increase, the need for maintenance will increase. Besides, since the program will be used in real life, the need for maintenance is important. Therefore, the maintenance becomes a design concern for this project and the aim is to keep the maintenance effort and cost at the lowest possible level.

2 High-level Software Architecture

2.1 Subsystem Decomposition

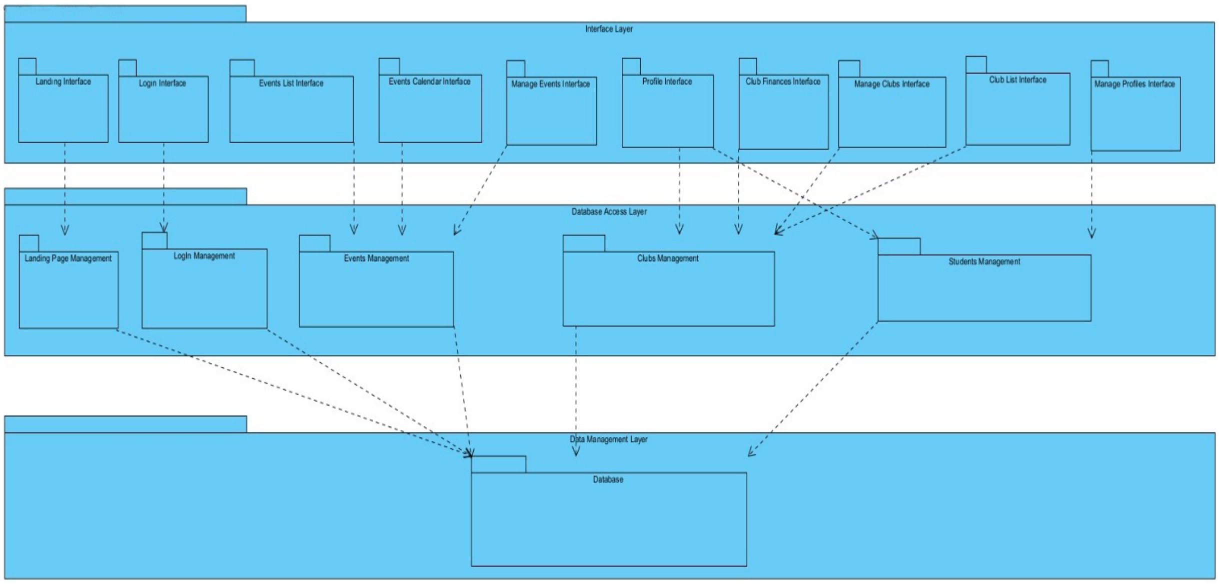

Figure 1. Subsystem Decomposition

The architecture of our system consists of 3 layers. The first layer, the Interface layer, covers the Front-end of our system. It includes all the important Interface pieces in packages.
The Database Access Layer still runs in the front-end code and is used to fetch and post relevant data from and to our Database. It is separated into packages in order to avoid a single package that has access to the whole database. Instead, every package has a specific purpose and accesses the relevant part of the Database.
Lastly, the Data Management layer consists of the Database that is controlled by the Database Access Layer.

2.2 Hardware/Software Mapping

The program doesn’t have any specific hardware requirement to function effectively. However, the primary features of a computer hardware, namely a keyboard, screen, mouse and a system sufficiently strong enough to run web browsers is needed. Additionally, the program can be used on tablets and mobile phones. For software, the program will be developed using Javascript ECMAScript 2015 and React Library version 17.0.2. As such, the following web browser versions and above are compatible with the program:

- Internet Explorer 10 and above
- Microsoft Edge 12 and above
- Mozilla Firefox 21 and above
- Google Chrome 23 and above
- Safari 6 and above

- Opera 15 and above
- Safari (iOS version) 6 and above
- Android browser version 4.4 and above
- Opera mobile version 64 and above
- Chrome for Android version 96 and above
- Firefox for Android version 94 and above

2.3 Persistent Data Management

To ensure the constant data flow within our app, we will use Firebase as our data management system. Firebase Realtime Databases provide JSON-formatted data that we will use to record our events, clubs and student profiles under different databases. The reason Firebase was chosen for this project is it’s great compatibility with Javascript libraries, fast request times, easy-to-implement functionalities and encrypted secure data transfer between our UI and databases. Firebase Authentication will be used to ensure secure, hashed LogIn. Also, instead of keeping passwords as plain-text, which is a really bad idea for security, Firebase Authentication makes sure even if someone can somehow fetch data from our database, they will not be able to view critical information such as passwords.

2.4 Access Control and Security

Due to its nature as a student club manager, the program enforces certain security precautions and access control. Login credentials, provided by the team, will be used to distinguish different types of users such as students, club profiles and SAC Admins. Functionalities aside, these user types also have different permissions. Students are not permitted to edit clubs, budgets or activities, they are only permitted to edit their profile, join or leave activities and/or clubs. Clubs have permissions allowing them to create activity or budget proposals that will be sent to the SAC Admin and can cancel approved or pending activities they have organized. Additionally, Student clubs can edit their profiles and are permitted to send general notification messages to their members, which is stored by the program. Finally, SAC Admins are given the greatest amount of access. They can edit their profile, are permitted to add or delete student clubs and can manage their budgets. They are allowed to accept or delete club events and can also delete approved club events.

We also take several security precautions in order to protect the personal information of our users. The Firebase Authentication Database and its built-in security features help us ensure safety for all sorts of authentication within the system.

2.5 Boundary Conditions

2.5.1 Initialization

The program is a web based application and therefore does not require installation. Most browsers can be used to access this site with an internet connection. Without an account, only public events can be seen on the landing page, along with a contact and a login page. After logging in, it is possible to list events, clubs and view the user's profile. The data displayed on these pages are retrieved from the database.

2.5.2 Termination

The program works separately, so the termination of one subsystem does not affect the entire application. Instead, that subsystem alone is terminated while other functionalities remain accessible. However, an admin can initiate a general termination, terminating all subsystems and therefore the entire application. Moreover, the program uses Firebase to save information and, in the case of a termination, the information on the system is saved before termination occurs.

2.5.2 Failure

The program uses the Firebase system for data storage and if the system fails it reacts by restarting itself. If this restart does not fix the failure, the developers are warned via a notification. For developer-related bugs, the system warns the developer before the changes are complete.

3 Low-level Design

3.1 Object Design Trade-Offs

3.1.1 Usability vs Functionality:

The program's aim is to make software as simple as possible so that every Bilkent University student can use this software. Though, making this program simple causes trouble about functionality since more operations makes the software more complex, then the usability will decrease. Therefore, some possible functionalities that can be used for this software are abandoned to ensure that the software will become more usable.

3.1.2 Usability vs Security:

The program offers some security properties to protect the data of the students, admin and clubs. However, the data security is limited at some point because making more secure software results in more complex software, so there is a conflict on the design goal. Since security is not included in the top design goals of the program, the idea of providing more security was given up to make the software usable.

3.1.3 Maintainability vs Performance:

One of the top design goals of the program is maintainability. The number of layers is enough to cover all parts of the software. It causes the loss of the performance. However, it grants great advantage to the maintainability of the program. Besides, though increasing maintainability causes performance loss, the performance can be abandoned to keep maintainability at its peak level.

3.2 Final Object Design

The final object design is given below.

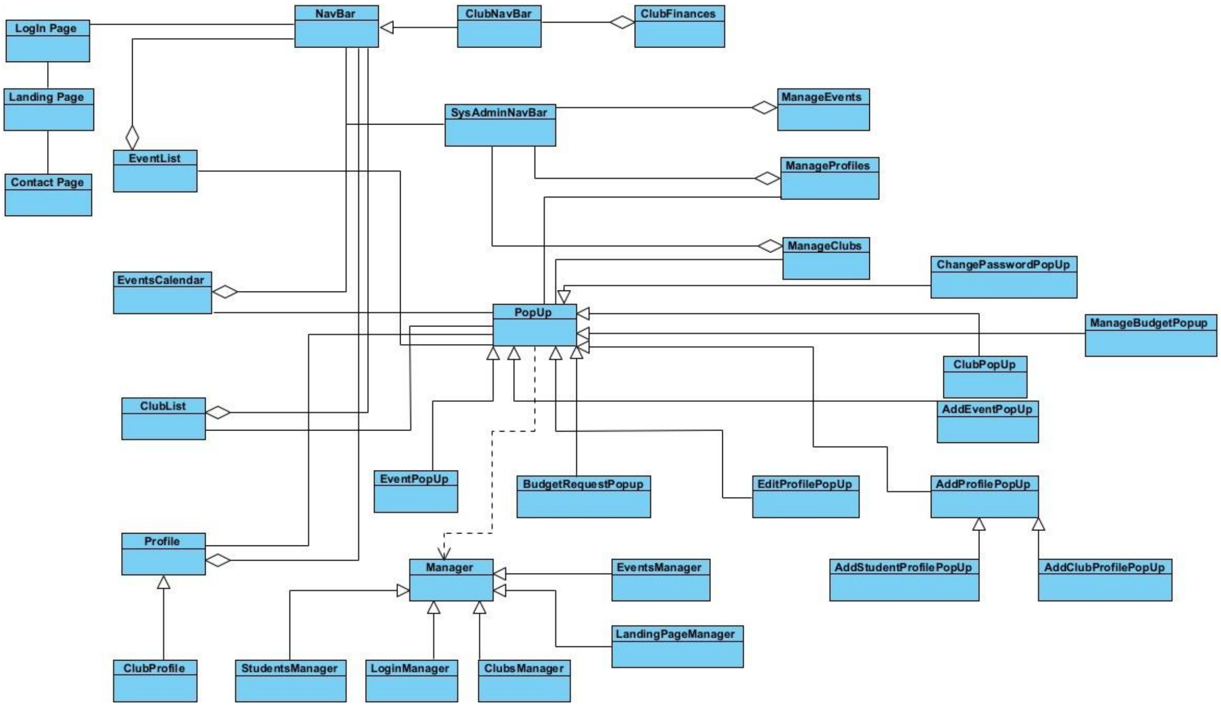

Figure 2. Final Object Design

3.3 Layers

3.3.1 Interface Layer

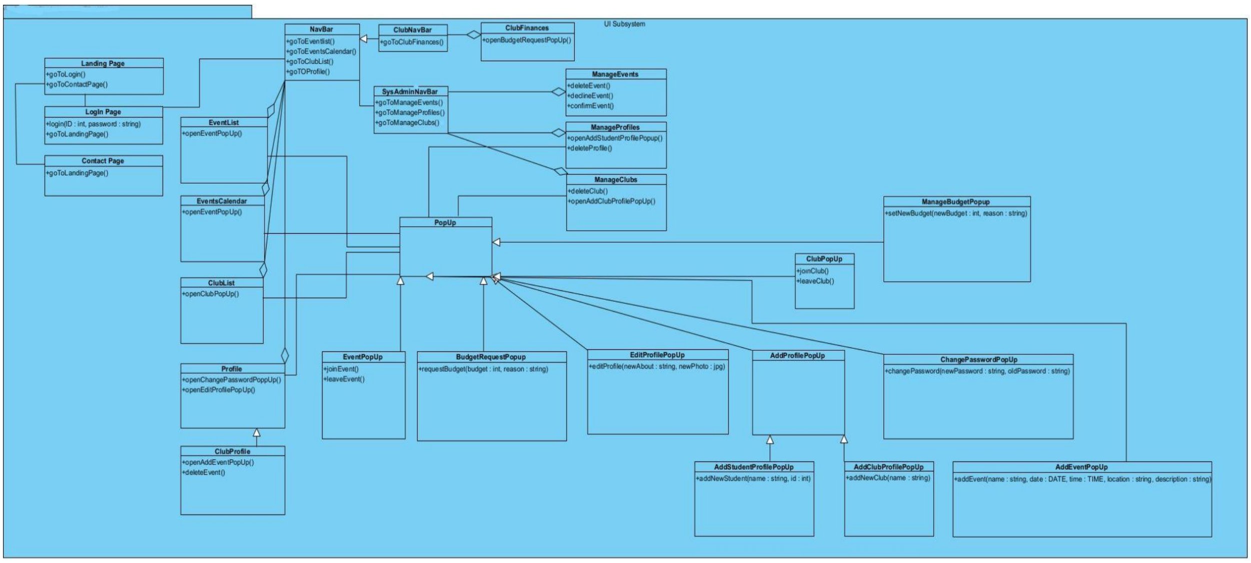

Figure 3: Interface Layer Class Diagram

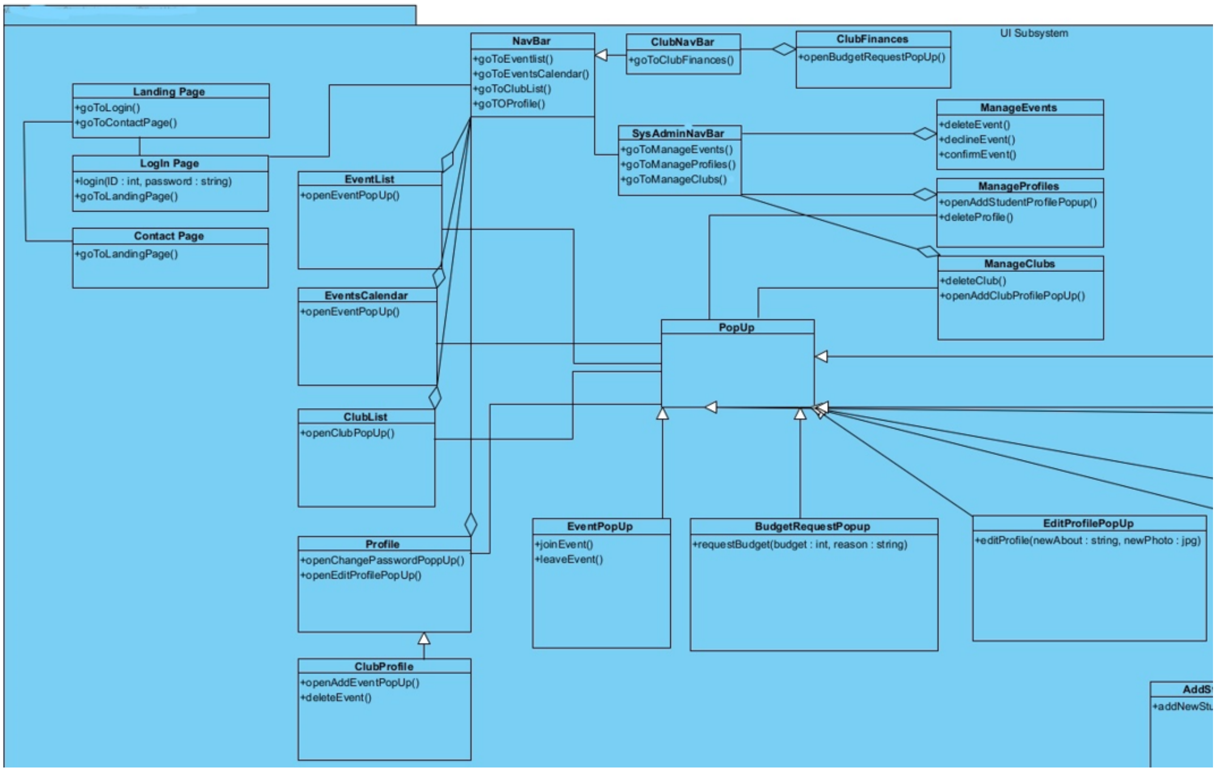

Figure 4: Left Side of the Interface Layer Class Diagram

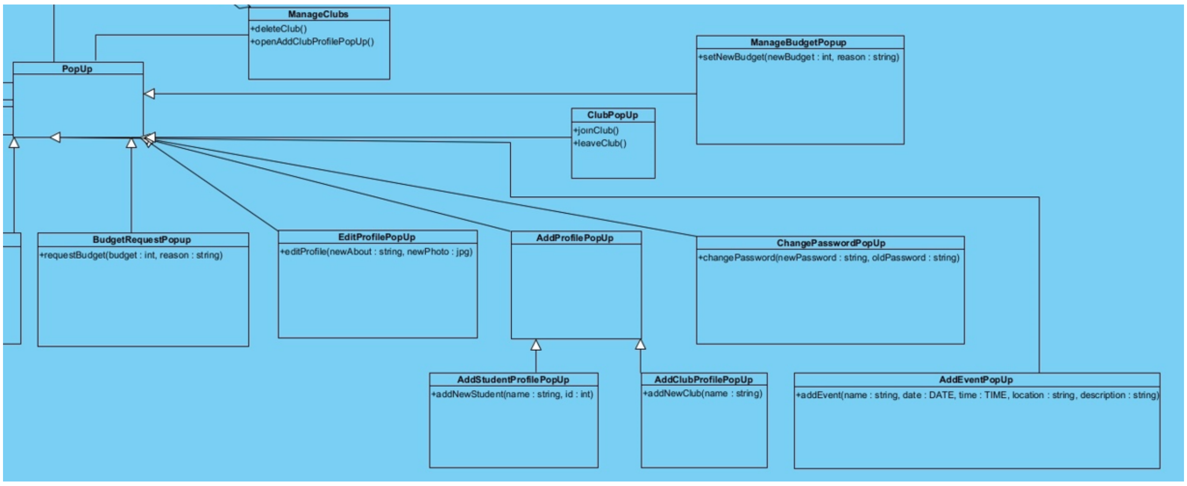
ManageClubs
+deleteClub()
+openAddClubProfilePopUp()
ManageBudgetPopup
+setNewBudget(newBudget : int, reason : string)
PopUp
ClubPopUp
+joinClub()
+leaveClub()
BudgetRequestPopup
+requestBudget(budget : int, reason : string)
EditProfilePopUp
+editProfile(newAbout : string, newPhoto : jpg)
AddProfilePopUp
ChangePasswordPopUp
+changePassword(newPassword : string, oldPassword : string)
AddStudentProfilePopUp
+addNewStudent(name : string, id : int)
AddClubProfilePopUp
+addNewClub(name : string)
AddEventPopUp
+addEvent(name : string, date : DATE, time : TIME, location : string, description : string)

Figure 5: Right Side of the Interface Layer Class Diagram

As noted before, the Interface layer consists of all the Front-end functionalities that users can communicate with. Each class in the layer are Javascript files that contain React Component Objects.

3.3.2 Database Access Layer

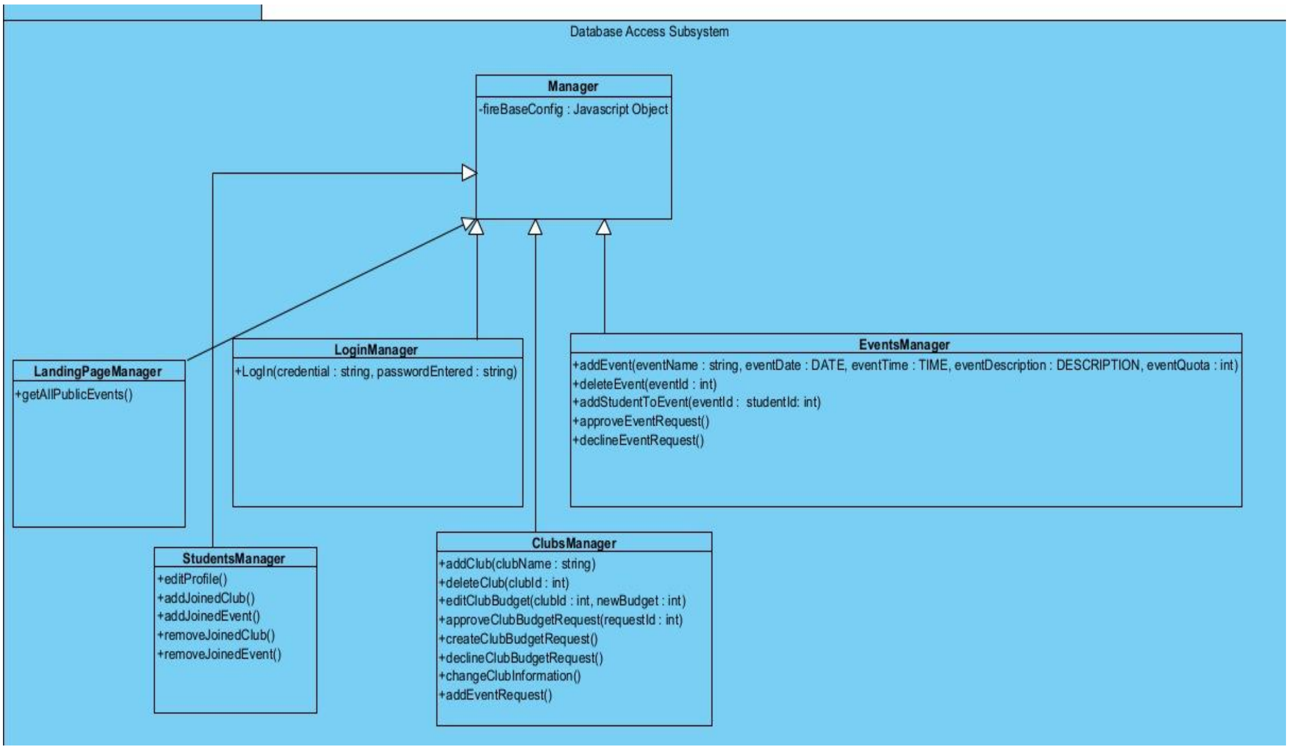
Database Access Subsystem
Manager
-fireBaseConfig : Javascript Object
LandingPageManager
+getAllPublicEvents()
LoginManager
+LogIn(credential : string, passwordEntered : string)
EventsManager
+addEvent(eventName : string, eventDate : DATE, eventTime : TIME, eventDescription : DESCRIPTION, eventQuota : int)
+deleteEvent(eventId : int)
+addStudentToEvent(eventId : studentId: int)
+approveEventRequest()
+declineEventRequest()
StudentsManager
+editProfile()
+addJoinedClub()
+addJoinedEvent()
+removeJoinedClub()
+removeJoinedEvent()
ClubsManager
+addClub(clubName : string)
+deleteClub(clubId : int)
+editClubBudget(clubId : int, newBudget : int)
+approveClubBudgetRequest(requestId : int)
+createClubBudgetRequest()
+declineClubBudgetRequest()
+changeClubInformation()
+addEventRequest()

Figure 6: Database Access Layer

The Database Access layer acts as a controller unit between the front-end and the firebase back-end. It includes the firebaseConfig file, which is used to store our database credentials.

3.4 Packages

3.4.1 Internal Packages

These are the packages specifically developed for this application.

3.4.1.1 LandingPageManagement

This package is responsible for the management of the public data displayed on the landing page.

3.4.1.2 Login Management

This package manages the login authorization data of the user.

3.4.1.3 Events Management

This package is responsible for management of the event data displayed on the Events List, Events Calendar and Manage Events interfaces.

3.4.1.4 Club Management

This package manages the data of Club displayed on interfaces related with the club which are Profile, Club Finances, Manage Clubs, Club List

3.4.1.5 Students Management

This package is responsible for the management of the student data displayed on the Manage Profiles and Profile interfaces.

3.4.2 External Library Packages

3.4.2.1 reactjs-popup

This library is used to help develop accurate and user-friendly customizable popups for the various uses it has within our app

3.4.2.2 react-calendar

This library will help us formulate the monthly event calendars for students and clubs.

3.4.2.3 styled-components

This React library will help us design the UI element stylings in classic CSS syntax, but with more functionalities and cleaner code.

4 Glossary and References

SAC - Student Activities Center

[1] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.