TEAM 10: Design Document
PROJECT NAME: ScheduMate
Project Coordinator: Adam D Rice

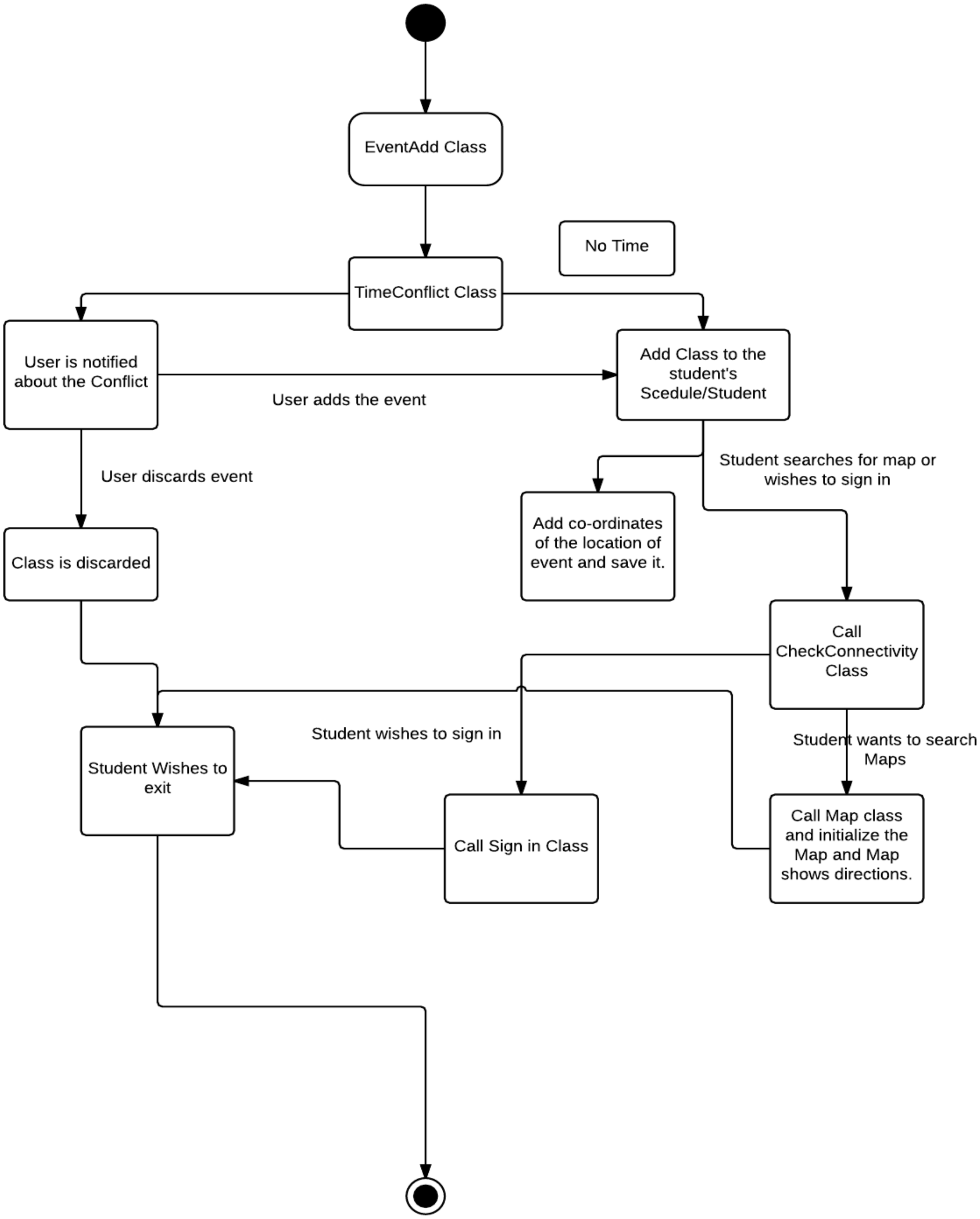
NIKHIL MEHTA
AISHWARYA AJAY
CHRISTOPHER HSIEH
ARITRA SAMANTA

Purpose

The purpose of this design document is to provide details about the Android application : **ScheduMate**. This document includes a summary of functional and nonfunctional design objectives, a design outline, design issues and design details.

ScheduMate is an Android application which will help students organize their daily schedule with ease. The application will create a planner which will help the student manage his/her time efficiently by consistently reminding the student of his/her daily activities.
It will also allow students find their way to classes with the help of campus maps and bus routes. The application will use efficient algorithms to calculate the best/shortest route to the class, so the student can reach his/her class on time.

Design Outline


EventAdd Class
No Time
TimeConflict Class
User is notified about the Conflict
User adds the event
Add Class to the student's Scedule/Student
User discards event
Student searches for map or wishes to sign in
Class is discarded
Add co-ordinates of the location of event and save it.
Call CheckConnectivity Class
Student wishes to sign in
Student wants to search Maps
Student Wishes to exit
Call Sign in Class
Call Map class and initialize the Map and Map shows directions.

The application only uses Google+ web service, and lets the user add the schedule online. That is why there is not a strong presence of the client-server Model. We will be using Queue and Tree Data Structures to set reminders for classes.

DESCRIPTION OF THE INTERACTIONS:

1. The student adds a class with the class details. As soon as the class details are added the TimeConflict class will check for time conflict among other classes.
2. If there are no conflicts, the class will be added to the Schedule. Note, initially Schedule will be empty.
3. If there are time conflicts, the user is prompted for addition of the class. If the user adds it, the class will be flagged and the user will be notified accordingly.
4. For non-conflicting classes, the reminders will be set along with other events, in the form of an alarm.
5. In the event of Map Usage, the Map class will be used, invoking CheckConnectivity Class and then connecting to the internet.
6. In the event of adding the schedule to Gmail, the the CheckConnectivity class is also called and then the user is asked to sign in and sync their schedule.

Design Issues

Non-Functional Requirements

1) Performance

Performance of an application is the primary concern for many developers and the consumers. The system as a whole should perform smoothly - without anything restraining it. Reaching as close as possible to a real time experience is a priority. This is to provide convenience for the user. For instance if the internet connection is slow, the connection between the server and client shouldn't disconnect. If this occured the client would become unresponsive, thus proving inconvenient. To handle such issues we will be giving the server most of the processing load. The client will be interactive while data is being continues to be synced asynchronously.

2) Usability

Easy navigation is extremely important as it affects the user's experience in almost all aspects. The application will adhere to the modern Google Application Development guidelines and we will make the application as user friendly as possible. The User Interface (UI) of the application will be laid out in a simple yet elegant manner. It will allow easy access to various menus and functions thus providing the best experience possible.

3) Reliability

Reliability is a main concern for any application. Emphasizing the user's need for convenience and efficiency , reliability plays a pivotal role in achieving these. For example, if a certain part of the system fails, the entire application will have to be restarted creating frustration as it wastes the user's time. Through proper programming practices we will balance performance and reliability so that we can maintain a reliable system without compromising performance.

4) Security

Securing the user's data will also be a primary concern as keeping an individual's data private is critical. No personal data from the user (e.g. login information, location, schedule) will be sent over the internet as plain text. Before transferring any kind of data, the application will ask the user for his/her permission. This will also enhance the usability and trustworthiness of the application. This is a key feature to ensure certain aspects of user information are kept confidential.

5) Battery Life
With smartphones being an essential part of everyday life, consuming large amounts of battery life can be detrimental to a user. The application will be designed to limit the background processes, running idle as much as possible. This will help ensure the application conserves as much battery life as possible. We hope that this will be one of the bigger plusses of our application as a product.

Functional Requirements

- **Programming Language**

Choosing a language is essential to any software development as each language has separate strengths and weaknesses.
Options:

1. Java
2. Objective C

Solution: Java
Reason: Each of the team members took CS 18000, which is the introductory course for all Computer Science majors. After taking CS 18000 and gaining first hand experience working on personal projects, everyone in our team is well versed with the Java Programming Language. Additionally, we are using Eclipse to write and test our code. The android toolkit provided by Google is widely considered as the best way to implement an Android application. Hence, we have opted to use it for our application.

- **Database Inclusion Issue**

Choosing where and how to store data is essential - the user will need secure, quick access to his/her information.
Options:

1. Including database in the phone itself.
2. Putting the database online.

Solution: Using the phone Database

Reason: We decided to use the phone's internal database to store user data to increase the performance of the application while providing added security. Although the application will use little more memory, it will be able to perform faster because there will be no transfer of data between the client-server over the internet.

- **Version Control**

A well organized, secure, and efficient method of working together is important to any team.

Options:

1. GitHub
2. SVN

Solution: GitHub

Reason: GitHub is one of the most popular repository systems on the market (if not the most popular). We chose it to synchronise our code/documents for many reasons. Our entire team has past experience with GitHub - there will be no learning curve among us. GitHub also provides us with a desktop application allowing each teammate to easily push code to a shared repository. This will be done without going through the hassle of logging in to the website. We did not choose centralised version control systems such as SVN because, although they are easy to use, the do not easily let the user manage servers and backup his code.

- **Maps**

Navigation to classes/buildings is one of the major features our application provides, so choosing the best method of implementation is key.

Options:

1. Google Maps
2. Bing Maps

Solution: Google Maps

Reason: As we are developing on the android platform, our first choice is to use Google Maps. Google Maps is the most widely used online maps in the world. There are many issues with Bing Maps. It is not as user friendly as Google Maps, and does not run nearly as efficient. It is also true that Bing Maps is not completely developed (and still full of bugs), Google Maps was easily the superior choice.

- **Offline Mode**

More options create a user friendly experience while providing utility.

Options:

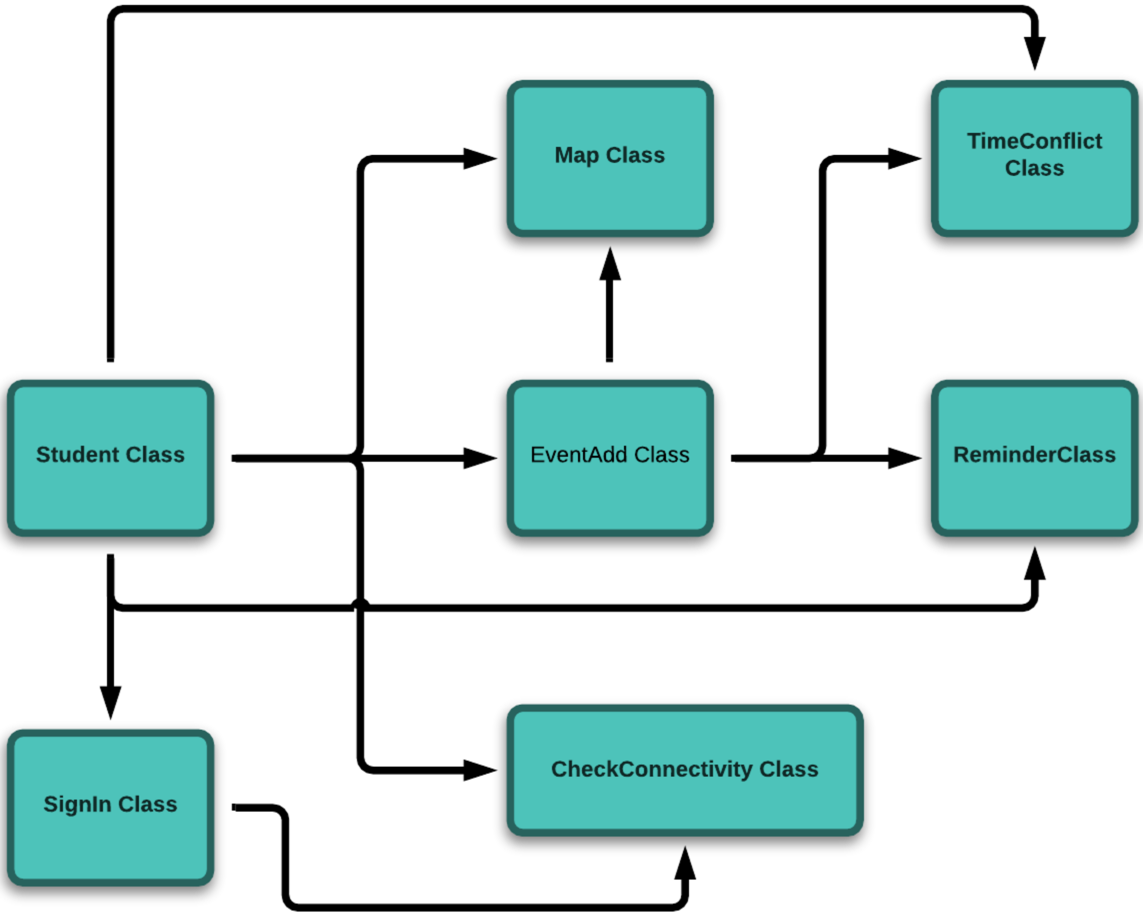
1. Allow the user to access maps when offline
2. The user doesn't have access to maps when there is no internet connection

Solution: Allow the user to access maps when offline

Reason: To make the application more user friendly and to increase its utility, we will allow the user to save a few maps which are used frequently so that he/she can find their way to class even connection to the internet is not available. This will also help the user save his/her cellular data when not connected to Wifi.

DESIGN DETAILS

CLASS LEVEL DIAGRAM AND INTERACTION OF CLASSES:



CLASS DESCRIPTIONS

1. **Student Class:** The student class will store the student's name, the classes the student is enrolled in, and other data stored for the particular student such as homework, events etc. Most importantly each event/assignment/class will have a time associated with it to

check for time conflicts upon addition of any other event/assignment/class. It will have a one-to-many relationship with all the other classes.

2. **TimeConflict Class:** The time conflict class will take in a time and check for conflicts and if there is any conflict, will notify the user about the same. If the user still wants to add the class, he/she can but it will be up to them to decide whether they want to manage it.

3. **CheckConnectivity Class:** This class checks for the connectivity of the phone to the internet. This class is used only when the user wants to send the data on his/her Gmail ID using Google+ sign. If the phone is not connected to the internet then, the user will be notified. the user may then connect to the internet, and let the app send an email. If not, then a reminder will be set securely on the app checking from time to time for an internet connection.

4. **Map Class:** The Map class will be called when the user wants to find a route to the class or the event. It will also help the user/student use the GPS facilities so that at any point the map can help the user go to a certain location pinned on his schedule. The Map class will also be connected to the Student class and store the default GPS locations from his home.

5. **Reminder Class:** This class will be interlinked with the Student class and will be responsible for reminder GUI creation for every event. It will have reminders with options to dismiss or attend an event.

6. **EventAdd Class:** This class will add an event to the Students Schedule. and take in the details of an event and check for conflicts by calling the TimeConflict class and the TimeConflict class will

take on from there, first checking for any clash/conflict and then letting the student add the data to his schedule.

7. **SignIn Class:** The sign-in class will be used so that the user can sign into Google+, and add the schedule online if he/she wants.