Cru iOS App: Software Architecture version 2.0

iCrew
Computer Science Department
California Polytechnic State University
San Luis Obispo, CA USA

December 2, 2015

Contents

Credits

Name	Date	Role	Version
Eric Tran	November 10, 2015	Document Owner and Creator of Sequence Diagram	1.0
Mariel Sanchez	November 10, 2015	Lead author of Introduction and Lead author of Overview and Creator of Class Diagram	1.0
Jordan Tang	November 10, 2015	Lead Author of Problem Descriptions and Creator of Component and Use Case Diagram	1.0
Daniel Lee	November 10, 2015	Creator of Activity Diagram	1.0
Tammy Kong	November 10, 2015	Creator of Deployment Diagram	1.0

Revision History

Name	Date	Reason for Changes	Version
Eric Tran	November 18, 2015	Changed layout of the document	2.0
Jordan Tang	November 18, 2015	Remade the deployment diagram	2.0
Tammy Kong	November 18, 2015	Added more sequence diagrams	2.0
Mariel Sanchez	November 18, 2015	Added more sequence diagrams	2.0
Daniel Lee	November 18, 2015	Activity diagram changes	2.0

1 Introduction

In this document, we will define the high level design and structure of the Cru App. We will provide a comprehensive architectural overview of the system using a number of views and diagrams to depict the different components of the software. These components include the ability to personalize the app to the user, look at upcoming events, find a ride to an event, and provide ways to further the users involvement with Cru. The intention is to clearly understand and convey the architectural decisions that have been made about the system to ensure that these functionalities are feasible.

2 Problem Description

Cru Central Coast is an organization that embodies multiple campuses and thousands of students. There are many events a groups that people can join in order to be involved with Cru. Due to the rapid expansion of the Cru body, it is becoming harder to organize people for events, ride sharing, and finding community groups. The application proposed will address these issues by having a ride-share feature, a list of upcoming events, and a search for finding community groups. Other content in the app will be resources from the Cru website such as articles and videos, a view of an academic calendar and map for the users campus, and applying for summer missions. The app will be available to current Cru members and new members alike.

3 Solution

3.1 Overview

In order to provide a solution to the problem, the Cru App will be developed with the mindset to be an all-in-one source for all Cru resources and information. Our system will be designed to be efficient and modular and our UI will be designed to provide easy and intuitive access to the important information.

With an increasing number of Cru members, it is becoming more difficult to ensure that all members are plugged in with what is going on. Our app will not only list upcoming events based on the users selected ministries and campus, but will also provide push notifications to these events as a quick reminder. With this feature, we are hoping that members of Cru will be more inclined to attend these events. To further influence users to attend Cru events, there will be an option to sync events to their own mobile calendar, providing multiple ways for members to be notified of upcoming events.

Another issue that was presented was the difficulty of setting up rides to events. We will provide a RideShare feature that, when enabled for an event, will automatically assign a number of passengers to a driver depending on factors from a questionnaire such as number of available seats, preferred leaving time, etc. This feature will relieve the

tedious and difficult task of manually going over these factors to pairing Cru members for rides.

The Cru App will also be a way for Cru members to deepen their involvement with Cru. We will design the system for members to join a Community Group based on their chosen ministries and a questionnaire in which the user will provide factors such as gender, time availabilities, etc. Users will also be able to see information about and join Ministry Teams through the app. Once users choose their Community Group or Ministry Team, leaders will be notified, providing an organized way for leaders to keep track of who shows interest in their group.

Other ways that the Cru App intends to provide Cru resources and bring together the Cru community is by including access to outside resources. This includes designing an embedded web app to display articles and guides. The app will also display embedded YouTube videos that are connected with different Ministry accounts.

In order to ensure that data displayed in the app is accurate and up to date we will design an architecture that will allow access to get and push data from the Cru database. We will also be implementing YouTube and Google APIs with our app to show the embedded videos and integrate with Crus Google Calendar. The details behind our design and implementation will be shown in the following sections of this document.

3.2 Components

The deployable key components in our design include the iOS device at which our application will be installed, and the accessing of Google and Youtube servers through their respective APIS for Cru videos and Google Calendar specifics. All of these components will be done on our end. The Cru MongoDB Database is a key component that has already been created and is currently managed by Cru. The Cru Web Client is another component in our design that currently exists and is using the Cru MongoDB Database.

3.2.1 Deployment Diagram

The Deployment Diagram displays the hardware for our system depicting a run-time configuration of processing nodes/hardware components. It effectively portrays the interaction of various systems such as one between an iOS device and the Cru server.

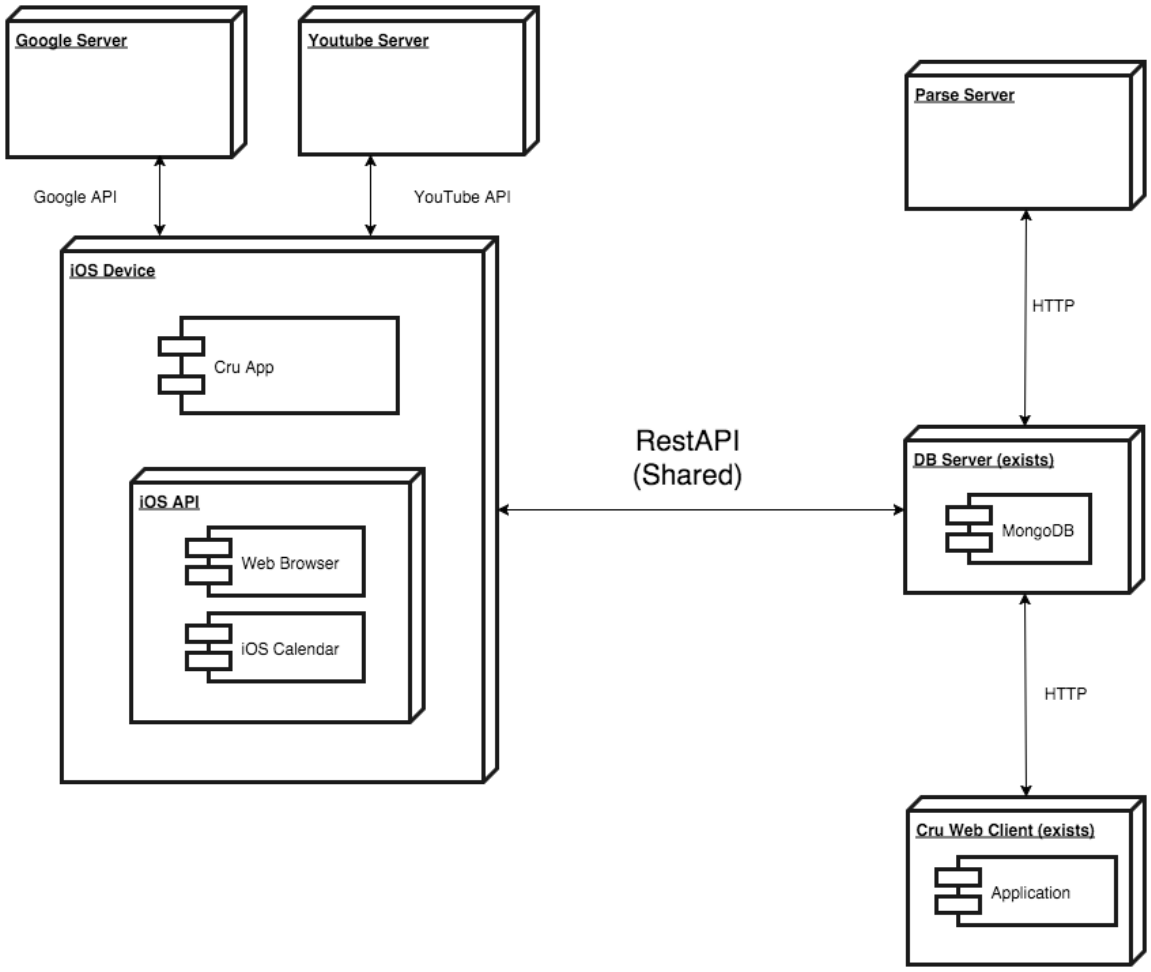

Figure 1: Deployment Diagram

3.3 Design

The following diagrams outline our architecture and design of the Cru iOS App.

3.3.1 Component Diagram

This component diagram displays each software component that will be required for our application. Most of our pieces should be separate software modules so it will be easy to plug them into other peoples application. The modules for are Events, Ride Share, Get Involved, Resources, and Summer Missions. The database will be another component, although that will already be provided to us.

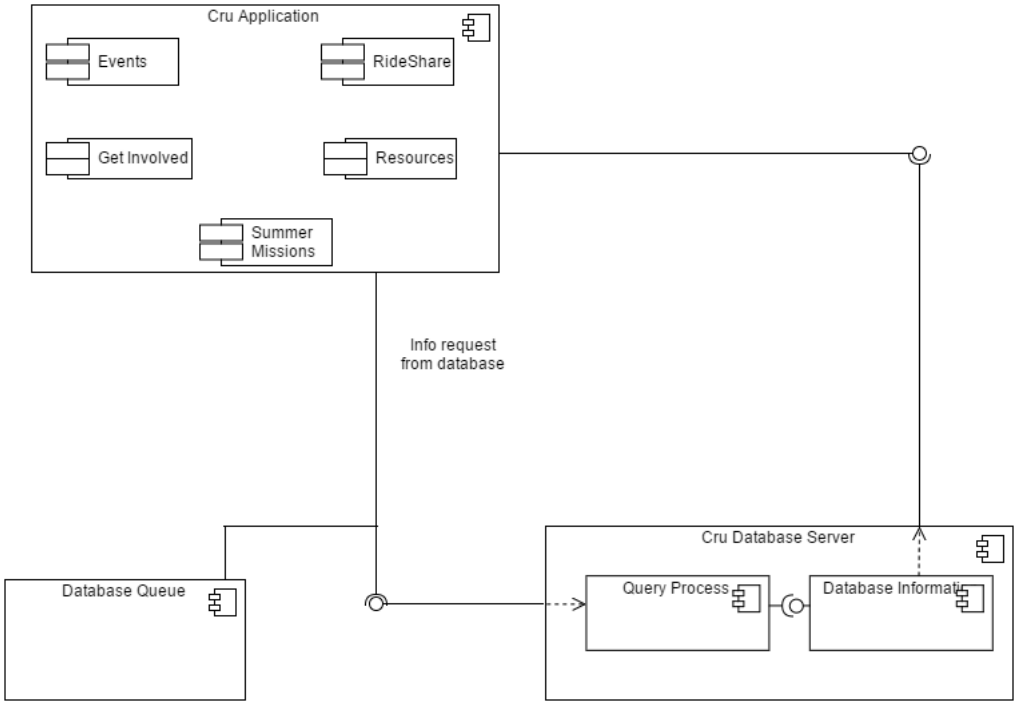

Figure 2: Component Diagram

3.3.2 Class Diagram

The class diagram below depicts the overview of our entire Cru App. In it, the dependencies of the classes within the system are shown.

There are multiple controllers to handle the different sections of the app: EventController, RideShareController, SummerMissionController, GetInvolvedController, Resource Controller, and MySchoolController. These are the menus of the app and almost all the other classes are under these controllers. There is a Person class that holds basic information and can be any user of the app as well as a Person pulled from the Cru database. There is also a UserProfile class that contains all the information that pertains to the device (i.e. the users ministries and communities). These classes are broken down even further into attributes and methods in the diagram.

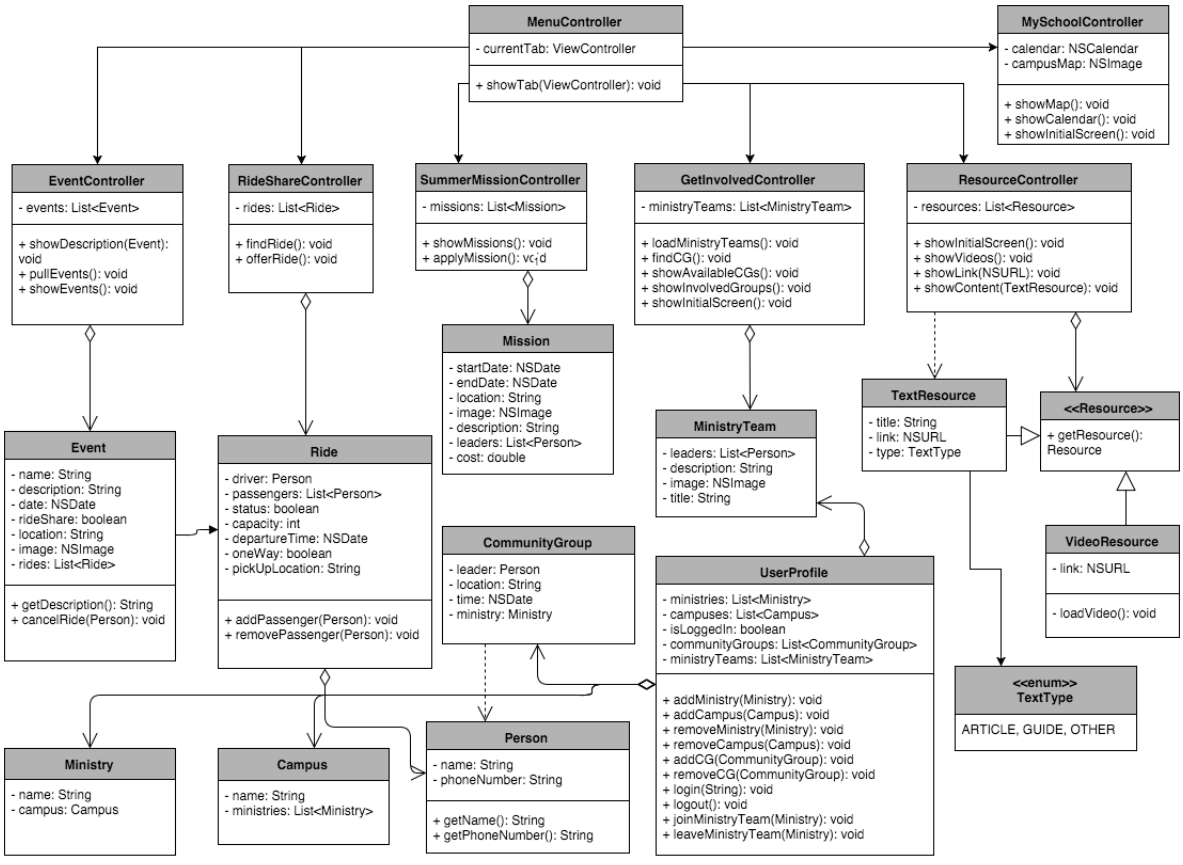

Figure 3: Class Diagram

3.3.3 Activity Diagram

The Activity Diagram represents a graphical workflow through the entire Cru app. Starting from the initial launch of the app, the user may step through activities (actions) laid out in each tab to accomplish various tasks such as joining a community group or ministry team. The main purpose of the diagram is to organize and evaluate possible actions and decisions.

The activity diagram is split into two diagrams for readability purposes.

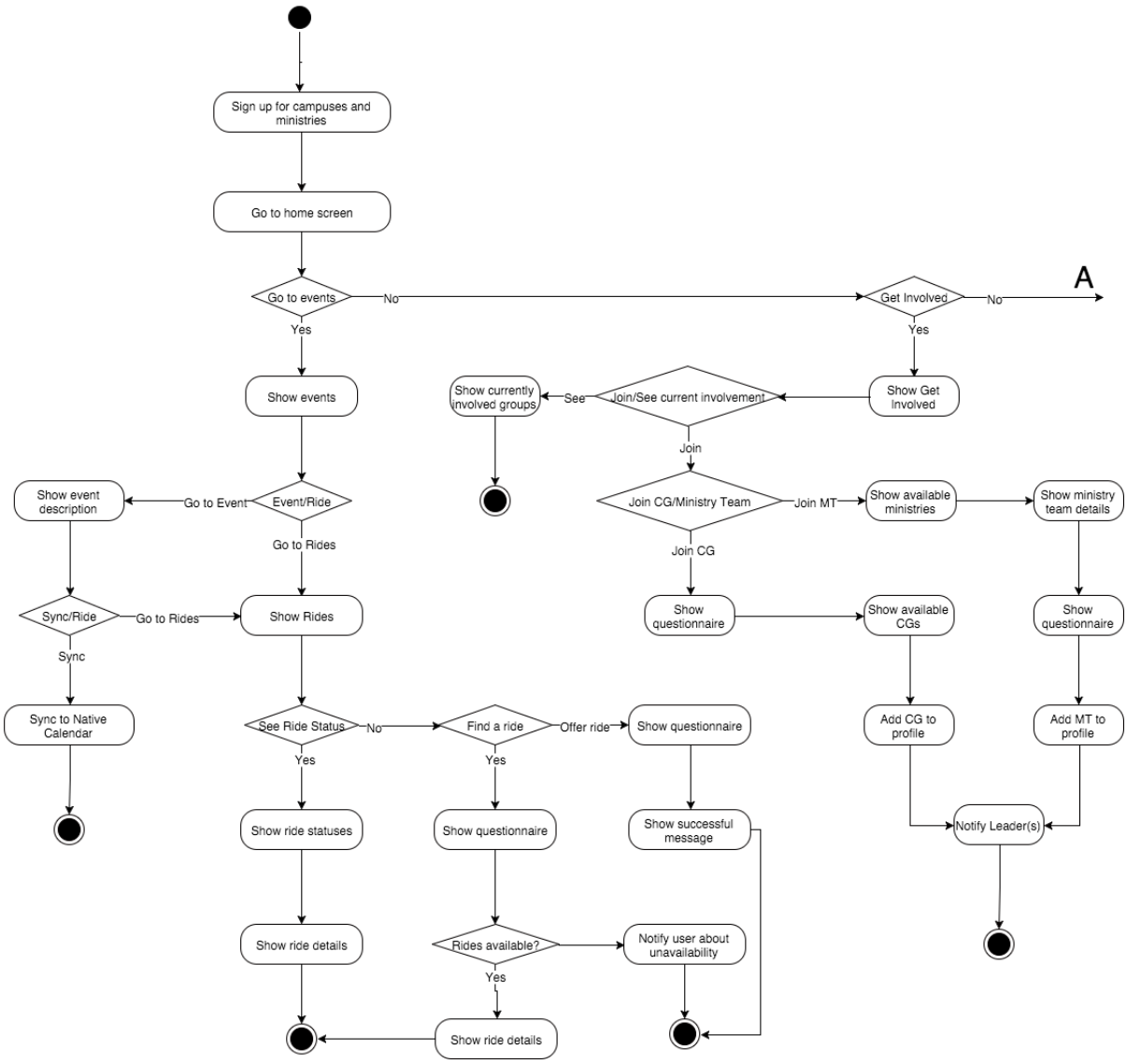

Figure 4: First Half of Activity Diagram

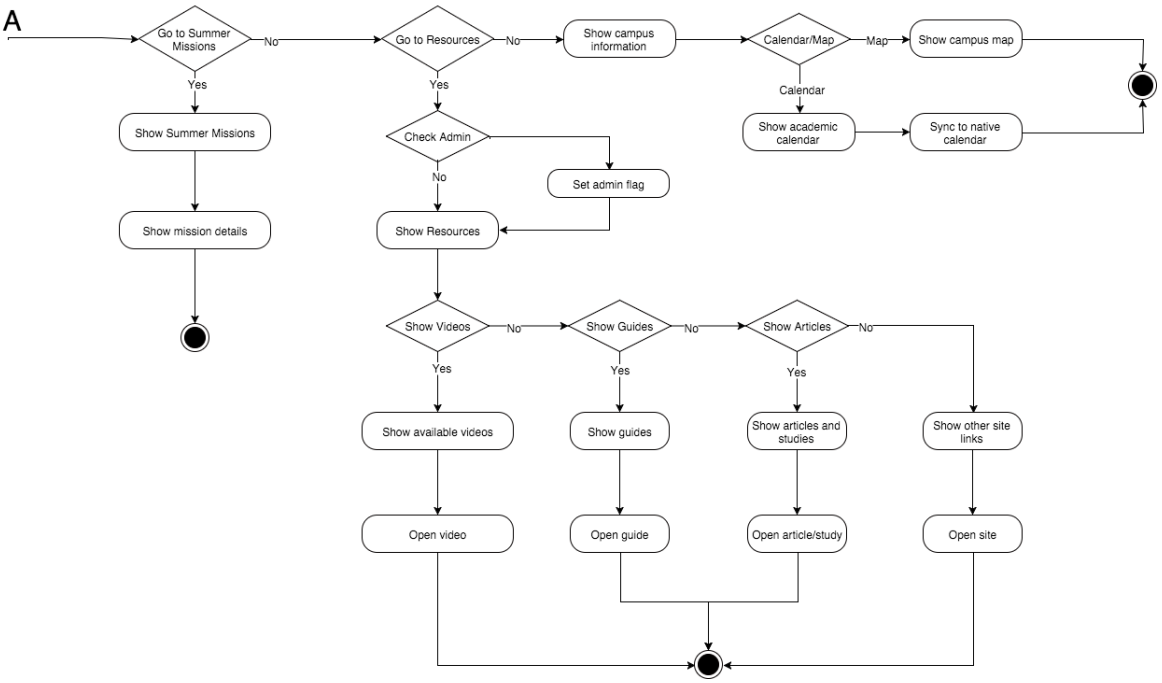

Figure 5: Second Half of Activity Diagram

3.3.4 Use Case Diagram

This Use Case diagram displays all the use cases and the actors involved with each use case. For a majority of the cases, the actor is the same - a member of Cru. There is another actor, a community group leader, who will have access to hidden resources within the app.

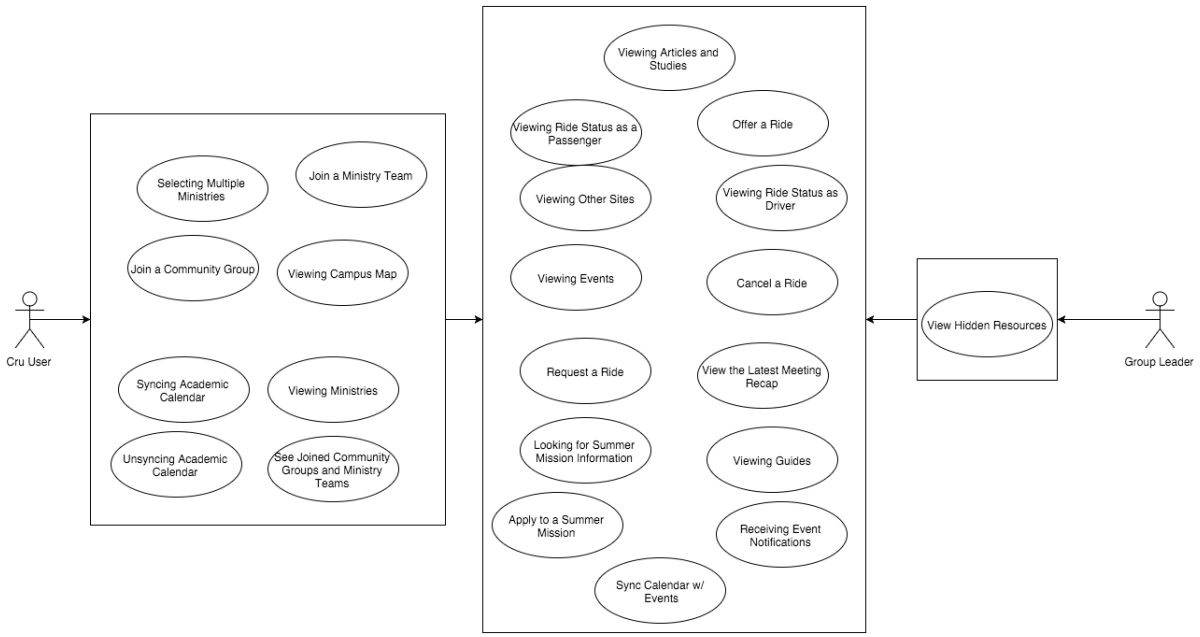

Figure 6: Use case diagram showing the different actors associated with all use cases

3.3.5 Sequence Diagram: Find a Ride

The following sequence diagram represents a specific use case: finding a ride as a user successfully. The diagram outlines the flow as a user would use the Cru App, and how each object is interacting with each other in our system.

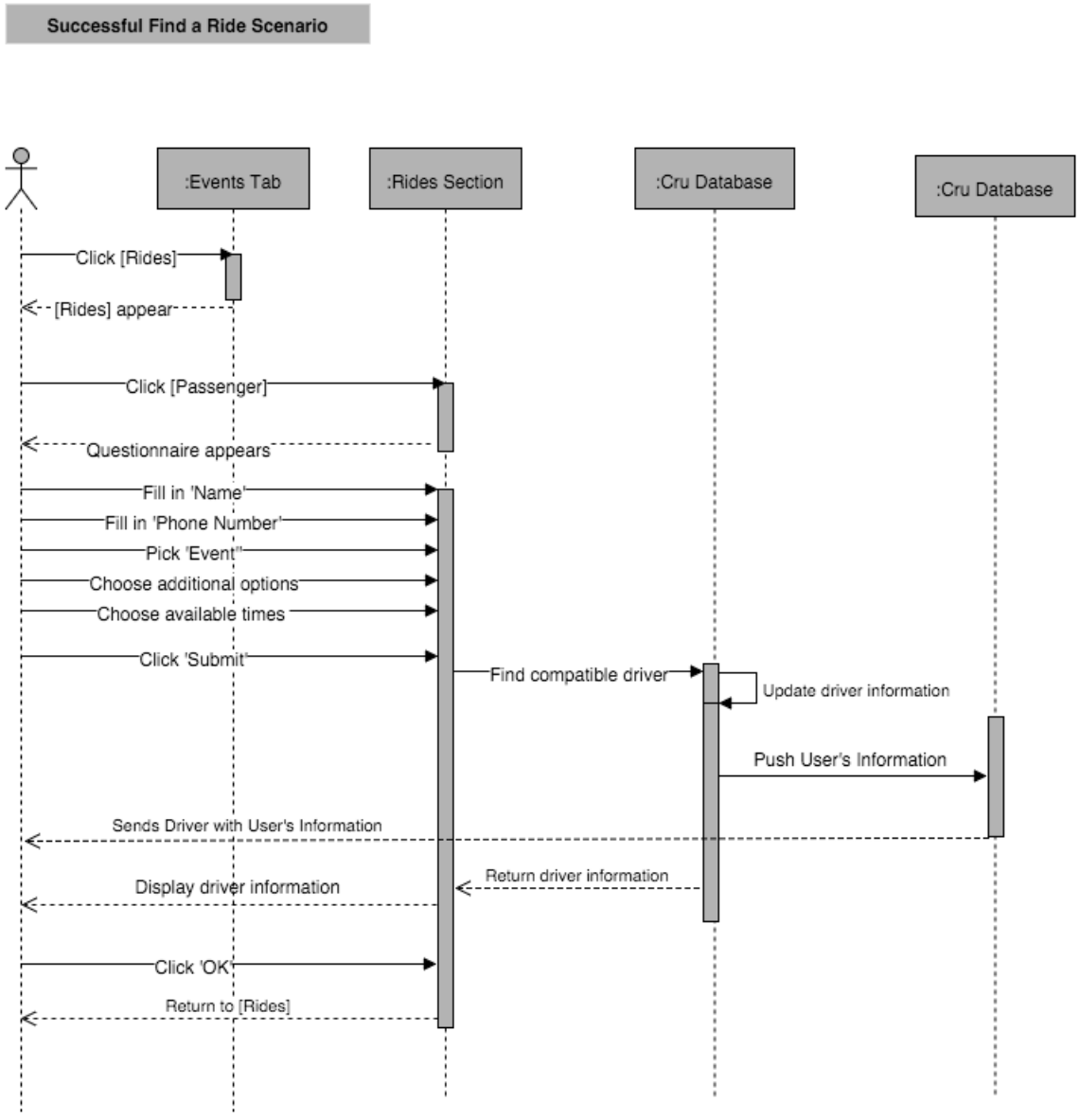

Figure 7: Sequence Diagram: Find a Ride

3.3.6 Sequence Diagram: Apply to a Summer Mission

The following sequence diagram represents a specific use case: applying to a summer mission from within the application. The diagram outlines the flow as a user would use the Cru App, and how each object is interacting with each other in our system.

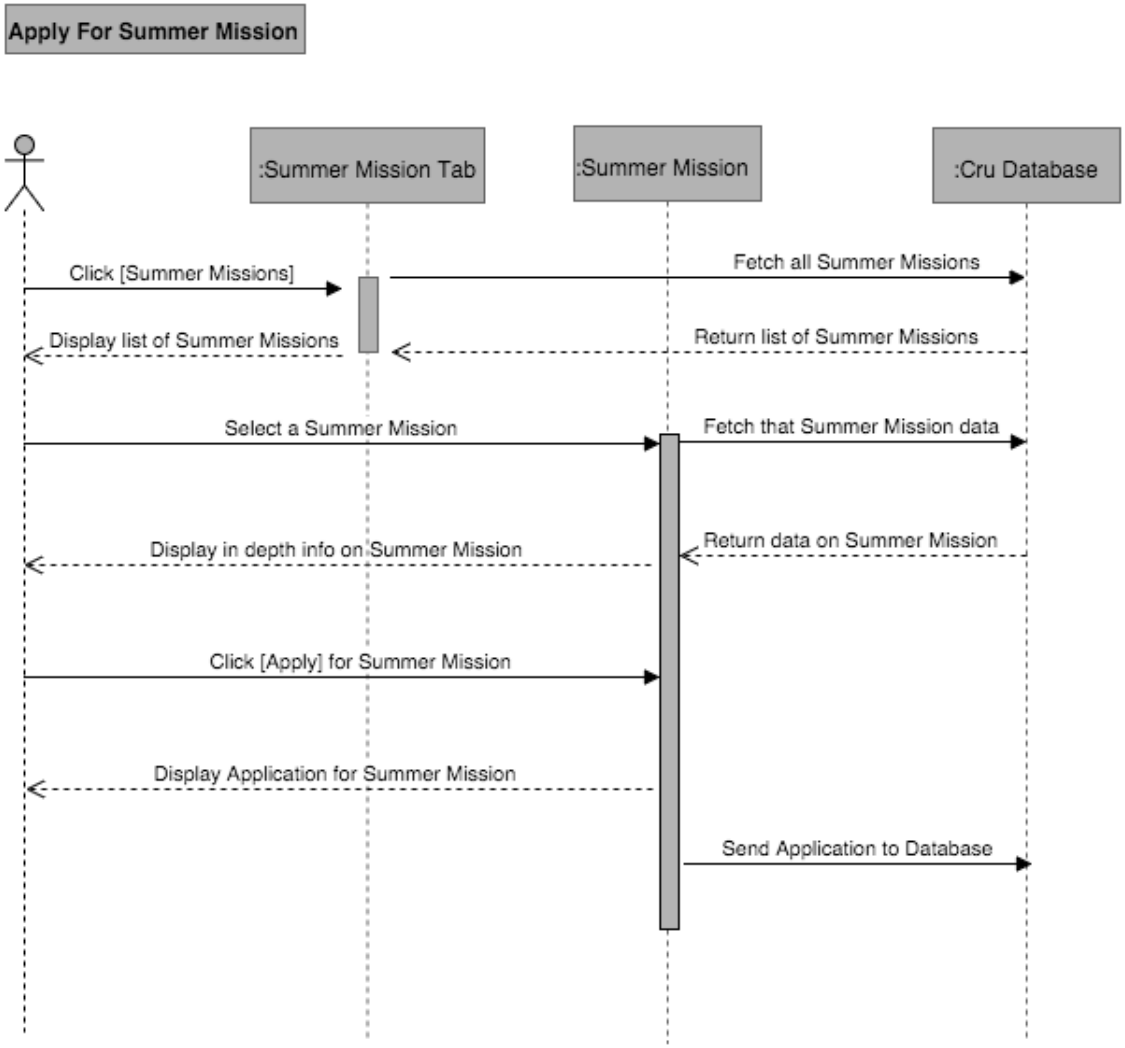

Figure 8: Sequence Diagram: Apply to a Summer Mission

3.3.7 Sequence Diagram: Joining a Community Group

The following sequence diagram represents a specific use case: joining a community group. The diagram outlines the flow as a user would use the Cru App, and how each object is interacting with each other in our system.

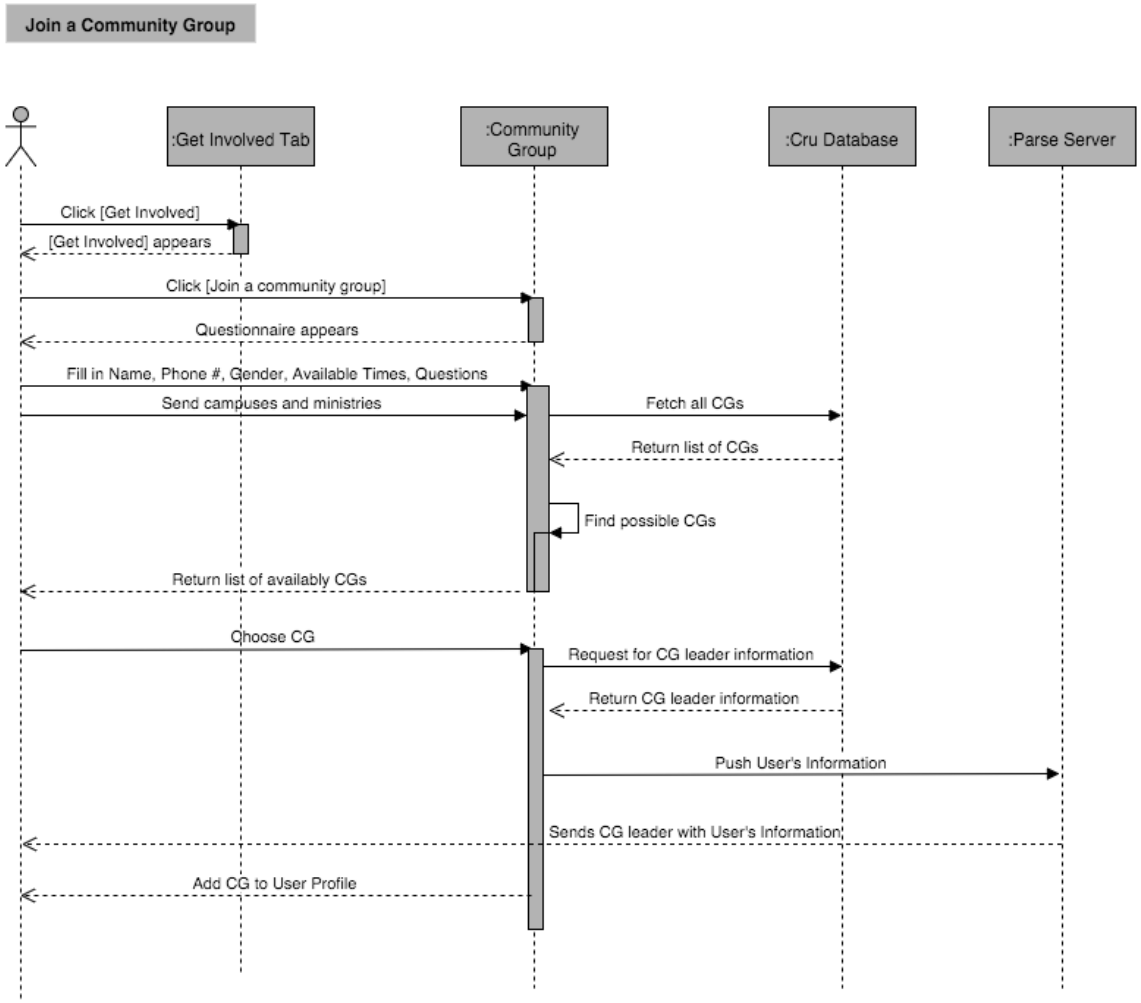

Figure 9: Sequence Diagram: Joining a Community Group

3.3.8 Sequence Diagram: Viewing a Youtube Video

The following sequence diagram represents a specific use case: accessing a Cru resource, specifically a Youtube video. The diagram outlines the flow as a user would use the Cru App to do so, and how each object is interacting with each other in our system.

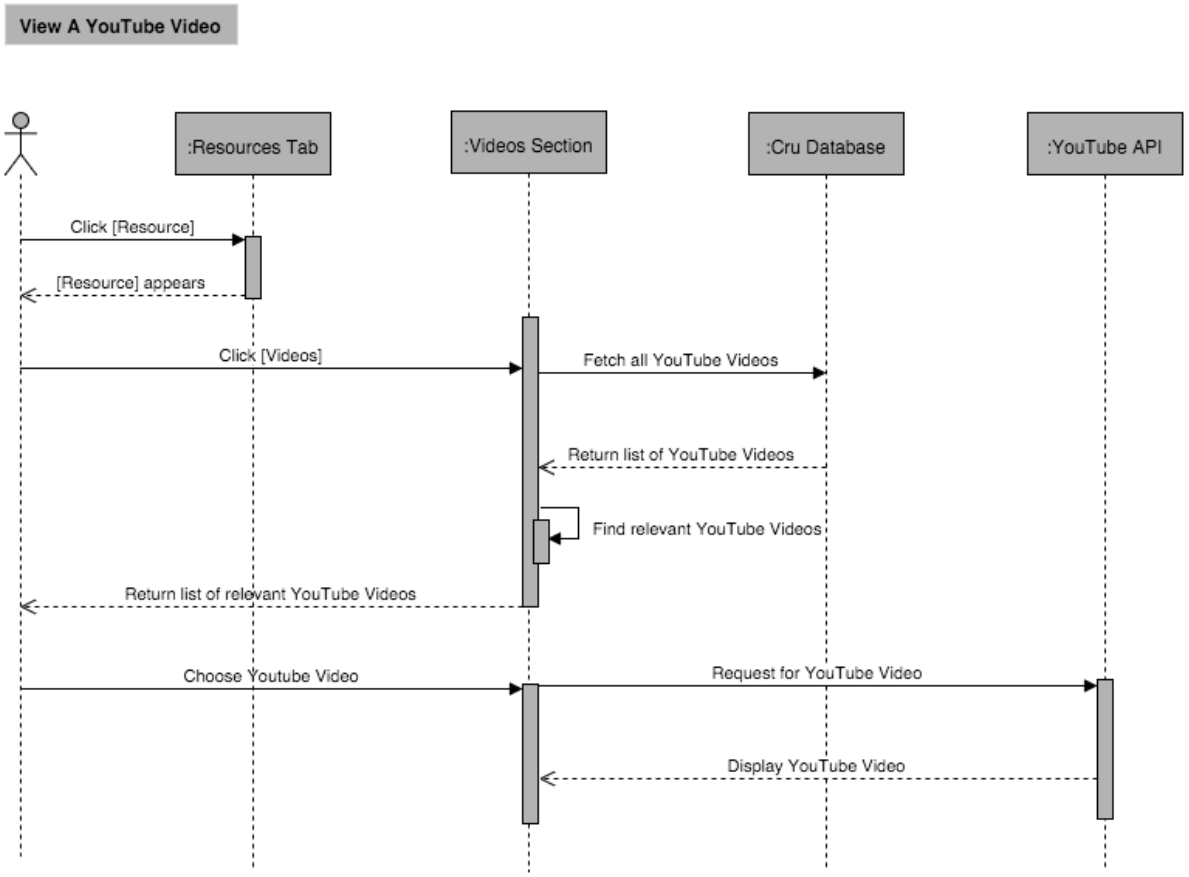

Figure 10: Sequence Diagram: Viewing a Youtube Video

4 Test

Our solution to the problem will be tested automatically with Jenkins, a continuous integration tool. We will be writing unit tests and acceptance tests in order to verify our solution works as intended.

5 Issues

A Glossary

B Issues List