

CRU App: Software Requirements Specification

Will Code for Food

Computer Science Department

California Polytechnic State University

San Luis Obispo, CA USA

October 27, 2015

Contents

[Revision History](#)

[Credits](#)

[1 Introduction](#)

[1.1 Purpose](#)

[1.2 Document Conventions](#)

[1.3 Intended Audience and Reading Suggestions](#)

[1.3.1 Developers](#)

[1.3.2 Mentor \(Kyle Fletcher\)](#)

[1.3.3 Supervisor \(David Janzen\)](#)

[1.4 Project Scope](#)

[1.5 References](#)

[2 Overall Description](#)

[2.1 Product Perspective](#)

[2.2 Product Features](#)

[2.3 User Classes and Characteristics](#)

[2.4 Operating Environment](#)

[2.5 Design and Implementation Constraints](#)

[2.6 User Documentation](#)

[2.7 Assumptions and Dependencies](#)

[2.8 Business Rules](#)

[3 Use Cases](#)

[3.1 Change Notification Settings](#)

[3.2 Access Community Leader Resources](#)

[3.3 Access a Ministry Resources](#)

[3.4 Join Community Group](#)

[3.5 Sign Up as an Event Driver](#)

[3.6 Finding a Ride](#)

[3.7 Viewing Existing Rides](#)

[3.8 Login](#)

[3.9 Join a Ministry](#)

[3.10 Change or Add Campuses](#)

[3.11 View Ministry Team Information](#)

[3.12 View and Apply to a Summer Mission](#)

[3.13 Join a Ministry Team](#)

[3.14 Add an Event to Calendar](#)

[3.15 Cancel a Ride](#)

[4 System Features](#)

[4.1 Subscribe to a Ministry](#)

[4.1.1 Description and Priority](#)

[4.1.2 Stimulus/Response Sequences](#)

4.1.3	Functional Requirements
4.2	Access Ministry Resources
4.2.1	Description and Priority
4.2.2	Stimulus/Response Sequences
4.2.3	Functional Requirements
4.3	Push Notifications
4.3.1	Description and Priority
4.3.2	Stimulus/Response Sequences
4.3.3	Functional Requirements
4.4	Access Leader Resources
4.4.1	Description and Priority
4.4.2	Stimulus/Response Sequences
4.5	Join a Community Group
4.5.1	Description and Priority
4.5.2	Stimulus/Response Sequences
4.5.3	Functional Requirements
4.6	Sign up as a Driver for an Event
4.6.1	Description and Priority
4.6.2	Stimulus/Response Sequences
4.6.3	Functional Requirements
4.7	Sign up for a Ride to an Event
4.7.1	Description and Priority
4.7.2	Stimulus/Response Sequences
4.7.3	Functional Requirements
4.8	Access Ministry Events
4.8.1	Description and Priority
4.8.2	Stimulus/Response Sequences
4.7.3	Functional Requirements
5	External Interface Requirements
5.1	User Interfaces
5.2	Hardware Interfaces
5.3	Software Interfaces
5.4	Communications Interfaces
6	Other Nonfunctional Requirements
6.1	Performance Requirements
6.2	Safety Requirements
6.3	Security Requirements
6.4	Software Quality Attributes
A	Glossary
B	Analysis Models
B.1	Context Diagram
C	Issues List

Revision History

Name	Date	Changes	Version
Gavin Scott	10/22/2015	Added “View Existing Rides” use case (3.7), added “Business Rules” (2.8), added “Communications Interfaces” (5.4), updated “Sign up as a Driver” use case (3.5).	1.2
Brian Quezada	10/25/2015	Added Access Ministry Events (4.8) Added View Ministry Team Information Use Case (3.11) Updated View Ministry Resources Use Case (3.3) Updated Access Community Leader Resources (3.2) Removed “Other Requirements” (7)	1.2
Kayla Carr	12/2/2015	Changed “Subscribe for Notifications” use case into “Change Notification Settings” use case Edited and added requirements in sections 4.1, 4.2, 4.3 Added summer mission and mission team requirements and sections Added some more requirements	1.2
Mallika Potter	10/26/2015	Added “Add Event to Calendar” use case Edited “Join Community Group” use case Edited Introduction Reformatted document	1.2
Mason Stevenson	10/25/2015	Updated Access Leader Resources Use Case (3.2) Updated Login Use Case (3.8) Updated Access Leader Resources Feature (4.4) Added Glossary (A) Added Analysis Models (B) Added Issues List (C)	1.2
Shelli Crispen	10/25/2015	Updated “Sign up for a Ride to an Event” Use Case (3.7) Added “Join a Ministry Team” Use Case (3.13) Updated section 4.7 Added “Cancel a Ride” Use Case (3.15)	1.2

Credits

Name	Date	Role	Version
Kayla Carr	10/11/2015	Functional Requirements (4.1, 4.2, 4.3, 4.6, UI1-3) Nonfunctional Requirements (PR1-2, SR1-3, SQA1-4) Subscribe to Notifications Use Case Push Notifications System Feature (4.3)	1.0
Shelli Crispen	10/11/2015	Overall Description (2.1, 2.2, 2.3) Sign Up for a Ride to an Event (3.7) Functional Requirements (4.7) Join a Ministry Team (3.13) Cancel a Ride (3.15)	1.0
Mallika Potter	10/13/15	Introduction (1) Join Community Group Use Case Join Community Group Feature (4.5)	1.0
Gavin Scott	10/13/2015	Sign up as a Driver Use Case Sign up as a Driver System Feature (4.6) Functional Requirements (4.6) Nonfunctional Requirements (PR-4, PR-5, SaR-1, SaR-2, SR-4, SQA-6) Product Perspective (2.1) Product Features (2.2)	1.0
Mason Stevenson	10/12/2015	Access Leader Resources System Feature (4.4) Access Community Leader Resources Use Case	1.0
Brian Quezada	10/13/2015	Introduction (1) View Article / Resources Use Case (3.3) Access Ministry Resources (4.2)	1.0

1 Introduction

1.1 Purpose

These requirements are for the Cru Mobile app v1.0. It is for the initial prototype of a Cru Android application for the Cru Central Coast organization.

1.2 Intended Audience and Reading Suggestions

1.2.1 Developers

The developers creating and maintaining this application will be using this document as a specification for the scope and purpose of the application. It will be used to define key features and limitations of development.

Important Sections:

1. Overall Description
2. Use cases.
3. System Features
4. External Interface Requirements.
5. Other Non-Functional Requirements

1.2.2 Mentor (Kyle Fletcher)

Our Mentor will use this document to ensure that our team is aware of all user needs for this project. Additionally, he shall ensure that the team is aware of Cru Central Coast technologies and business rules.

Suggested Sequence:

1. Overall Description.
2. Use cases.
3. System Features.
4. Other Nonfunctional Requirements.
5. External Interface Requirements.

1.2.3 Supervisor (David Janzen)

Our supervisor will use this document to remain informed of our team's decisions and progress.

Suggested Sequence:

1. Overall Description.
2. Use cases.
3. System Features.
4. Other Nonfunctional Requirements.
5. External Interface Requirements.

1.3 Project Scope

This project will create a native Android application which people can use to get information about Cru ministries, connect to Cru community groups and ministries, and share rides to Cru events.

1.4 References

These resources may be useful alongside this document.

1. [Will Code For Food Vision and Scope Document](#)
2. [Horizontal Prototype](#)

2 Overall Description

2.1 Product Perspective

The app is a new mobile interface for CRU Central Coast and it syncs with their existing databases. It's a new system but it will have the same look and feel as the other CRU Central Coast resources so that existing CRU Central Coast members can pick it up easily. It is an alternative to some of the cumbersome activities that have typically been done manually, such as ride-sharing, event planning and sending information to the CRU members. Figure D-1 illustrates the external entities and system interfaces for the initial release (figure still to be made).

2.2 Product Features

The following is a list of the major intended features for CRU mobile. They represent high level goals for the app.

FE-1	Subscribe to a ministry.
FE-2	Access ministry resources.
FE-3	Push notifications for ministry events.
FE-4	Join community group based on user's personal information.
FE-5	Offer rides to events.
FE-6	Find and join offered rides.
FE-7	Access community leader resources.
FE-8	Access ministry events.
FE-9	Join a summer mission.
FE-10	Join a ministry team.

2.3 User Classes and Characteristics

User Class	Description
------------	-------------

First Time Member	Downloads the app and comes to the opening page where they can sign up for the specific ministry and campus which they are looking to join. The app gives them the news feed and events data for that ministry and if they choose they can sign up to be apart of a community group.
Group Leader	Uses the app as a resource for all the campuses information and events as well as for extra resources only for group leaders via a login.
Regular Members	Uses the app as a resource for all the events, sermons, and give them the ability to drive to an event or get a ride for the event. Members can find information to help out and get involved in summer missions and other Cru programs.

2.4 Operating Environment

OE-1	The app shall run natively on Android v4.4 - 5.1.
OE-2	The app shall pull data from the existing CRU database.
OE-3	The app shall comply with CRU visual and coding standards.
OE-4	The app shall function on multiple screen sizes.

2.5 Design and Implementation Constraints

User experience: No experience with CRU is necessary, new and potential members should be able to easily understand the main functionality of the app.

2.6 User Documentation

Currently, there are no planned documents for user instruction.

2.7 Assumptions and Dependencies

Cru databases will be available for use by our application.

Design of the use shall adhere to the Cru design guidelines.

2.8 Business Rules

Business Rule	Description
BR-1	Community Leaders must be authenticated using CRU's authentication service, using a username and password assigned to them by CRU administration.

BR-2	Users' personal information will not be stored in the database except in the case of community leaders and users using ride-sharing.
BR-3	Users using ride sharing will not have their personal information stored in the database after the event(s) they are attending have ended.
BR-4	Users offering a ride will have their phone numbers validated against a collection of Cru member phone numbers to ensure the driver is a member of Cru.

3 Use Cases

3.1 Change Notification Settings

Use Case ID	1
Use Case Name	Change Notification Settings
Created By	Kayla Carr
Last Updated By	Kayla Carr
Date Created	10/11/2015
Date Last Updated	10/24/2015
Actors	Existing members of Cru Community leaders and staff of Cru Potential new members of Cru
Description	Allows the user to specify whether they want to receive notifications for events, ministries, or ride sharing.
Preconditions	1. User has joined a ministry or ministries 2. User has joined a campus
Postconditions	1. User has specified whether or not they want notifications. 2. System has recorded decision and will or will not send notifications.
Normal Flow	1.0 Set notifications 1. User accesses the "settings" section of the application. 2. User specifies whether they want to receive notifications for events. 3. User specifies whether they want to receive notifications about their ministry. 4. User specifies whether they want to receive notifications about their rides. 5. User saves their settings. 6. System updates the user's settings.

Alternative Flows	None
Exceptions	None
Includes:	None
Priority	Medium to Low
Special Requirements	None
Frequency of use:	Low usage
Business Rules:	None
Assumptions:	None
Notes and Issues:	None

3.2 Access Community Leader Resources

Use Case ID:	2
Use Case Name:	Access Community Leader Resources
Created by:	Mason Stevenson
Last Updated by:	Brian Quezada
Date Last Updated:	10-25-2015
Actors:	Community Group Leader
Description:	In addition to the regular student resources in the resources section, community group leaders will have access to an additional 50-60 articles organized by topic. If the user is logged in to the app as a community leader, the community leader resources section will appear in a subsection separate from student resources, but still within the resources section of the app.
Preconditions:	1. The user is logged in as a community leader.
Postconditions:	1. The community group leader has successfully accessed resources within the community leader resources subsection.
Normal Flow:	1.1 Access Community Leader Resources 1. The community group leader opens the app. 2. The community group leader navigates to the resources section of the

	app. 3. The community group leader selects a resource from the community leader resources subsection. 4. The resource is displayed.
Alternative Flows:	1.1 Not logged in (after 2) a. The community group leader selects the leader resources subsection. b. The user is prompted to login. c. The user logs in. d. Return to step 3. 1.2 User logs out. (after 2) a. The community leader resources subsection prompts for a login.
Exceptions:	None
Includes:	None
Priority:	High
Frequency of use:	Low usage
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.3 Access An Article From A Ministry

Use Case ID	3
Use Case Name	Access an article from a ministry
Created By	Brian Quezada
Last Updated By	Brian Quezada
Date Created	10-12-2015
Date Last Updated	10-25-2015
Actors	Cru Member
Description	A member can view resources related to ministries or campuses they have

	selected. Videos and articles are available for a member to view.
Preconditions	1. Member should be subscribed to the ministry
Postconditions	None
Normal Flow	<ol style="list-style-type: none"> 1. Go to resources 2. Filter or select articles. 3. Look at or click on an article to read it. 4. The article will be displayed with images and text fitting naturally inside the mobile screen size.
Alternative Flows	<ol style="list-style-type: none"> 1. Search for ministry related content. <ol style="list-style-type: none"> 1.1. Go to the search page 1.2. Enter keyword relating to that ministry 1.3. Select article 1.4. View article 2. View a ministry video <ol style="list-style-type: none"> 2.1. Go to resources 2.2. Filter or select videos 2.3. Select a video 2.4. The video will appear in app or redirect you to Youtube to view the video. 3. View community leader resources <ol style="list-style-type: none"> 3.1. See use case 2 (section 3.2)
Exceptions	None
Priority	High
Special Requirements	None
Assumptions	None

3.4 Join Community Group

Use Case ID	4
Use Case Name	Join Community Group
Created By	Mallika Potter
Last Updated By	Mallika Potter
Date Created	10/13/2015

Date Last Updated	10/13/2015
Actors	Cru Member
Description	Cru Member wants to register for a community group. This will involve receiving community group leader contact info and submitting their contact info.
Preconditions	1. Member has selected a campus.
Postconditions	None
Normal Flow	<ol style="list-style-type: none"> 1. Member selects "Get Involved" 2. Member selects "Community Group" 3. Member fills out questionnaire. 4. System displays available community groups. 5. Member selects community group.. 6. Member inputs contact info to be sent to community group leader. 7. Member saves community group leader contact info.
Alternative Flows	None
Exceptions	None
Includes:	None
Priority	High
Special Requirements	None
Frequency of use:	Medium usage
Business Rules:	None
Assumptions:	Cru user is subscribed to at least one ministry.
Notes and Issues:	None

3.5 Sign Up as an Event Driver

Use Case ID:	5
Use Case Name:	Sign Up as an Event Driver
Created By:	Gavin Scott
Last Updated By:	Gavin Scott

Date Created:	October 13, 2015
Date Last Updated:	October 22, 2015
Actors:	Standard CRU User
Description:	A user has decided that they wish to attend an event, and they want to offer rides to the some other members of CRU.
Preconditions:	<ol style="list-style-type: none"> 1. The user is viewing the event page for the event to which they want to drive.
Postconditions:	<ol style="list-style-type: none"> 1. The user has been registered as a driver for that event and will be visible to other members inride finding section of that event. 2. The driver will receive notifications when a user has signed up for a ride, and when all available seats in the car have been claimed.
Normal Flow:	<p>1.0 Sign up as an event driver</p> <ol style="list-style-type: none"> 1. The user navigates to the ride-sharing page and expands the desired event. 2. The user selects the option to drive to the event. 3. The system checks the driver's name and phone number against the database to verify that they are a member of CRU. 4. A form appears asking how many seats are available in the user's car, when they want to leave for the event, whether or not they can drive both ways, etc. 5. The user fills out the information correctly and submit the form. 6. A confirmation message appears, telling the user that they will receive notifications when another CRU member claims a seat. The system sends the user's information, including name and number, to the database. 7. The user acknowledges the confirmation. 8. The screen returns to the list of events that have ride-sharing enabled.
Alternative Flows:	<p>1.1 The user decides not to drive before they submit</p> <ol style="list-style-type: none"> 1. The user cancels the process before submitting their information and the entered data is thrown out. 2. The screen returns to the event page.
	<p>1.2 The user enters an invalid information into the form</p> <ol style="list-style-type: none"> 1. The system displays an error message when the user submits the information, informing them of their error and not sending data to the database. 2. Return to step 2.
	<p>1.3 The user is not a registered member of CRU</p> <ol style="list-style-type: none"> 1. Start at step 1.0.3. 2. The system displays an error saying that the user is not registered with

	CRU and cannot sign up as a driver.
Exceptions:	1.0.E.1 Failure to save data to the database <ol style="list-style-type: none"> 1. System displays an error message explaining the problem. 2. Return to step 2, with previously entered data still filled out. 3. The user can continue to try to submit, or can cancel to try later or contact a system administrator.
Includes:	None
Priority:	High
Frequency of Use:	Low usage per user, usually only once per event they attend.
Business Roles:	TBD
Special Requirements:	TBD
Assumptions:	The user will be honest about the number of available seats in their car. The user can legally drive themselves and other people to the event.
Notes and Issues:	<ol style="list-style-type: none"> 1. Will the system ask for information such as make, model, and color of the car to make it easier to find, or should that information be communicated once the driver has been put in contact with the people getting rides?

3.6 Finding a Ride

Use Case ID	6
Use Case Name	Finding a Ride
Created By	Shelli Crispen
Last Updated By	Gavin Scott
Date Created	October 13, 2015
Date Last Updated	October 26, 2015
Actors	Standard CRU User
Description	A CRU event is about to happen and members have the opportunity to use the rideshare feature of the new CRU SLO app. A member places themselves in a car through the app and it alerts the driver to how many people they are driving.
Preconditions	<ol style="list-style-type: none"> 1. Drivers must be verified members of CRU

	2. The event in question must have ride-sharing enabled
Postconditions	1. The user has been registered by the system as a rider in a specific car for the event, or the system will notify the rider when a ride becomes available.
Normal Flow	1.0 Sign up as an event rider from main page <ol style="list-style-type: none"> The user navigates to the ride-sharing tab The user selects an event, then that they need a ride The user enters the required information, including desired departure time, one-way vs. two-way trip, and where they want to leave from, and submits the form. The system displays several options for rides that they can search through, sorted based on the closeness of the ride to the user's requested time of departure. They select a ride. The system adds the user to the car, notifying the driver. The driver's and user's contact information is exchanged. The system returns to the ride-sharing tab.
Alternative Flows	1.1 Sign up as an event rider from event page <ol style="list-style-type: none"> Looking through the "Events" tab a member finds an event they would like to attend. The user selects the ride-sharing option for that event. Begin at step 1.0.3.
	1.2 There are no available rides for the event <ol style="list-style-type: none"> Start at step 1.0.3. The user submits their information. The system displays a message explaining the lack of rides, and that the user will be notified if any become available. The system returns to the ride-sharing tab.
Exceptions	1.0. Car fills up with riders <ol style="list-style-type: none"> System displays "Car is currently full." and send a message to the driver saying "Your car is currently full."
Priority	High
Special Requirements	TBD
Assumptions	Ability to find a ride to an event for CRU.

3.7 Viewing Existing Rides

Use Case ID	7
-------------	---

Use Case Name	Viewing Existing Rides
Created By	Gavin Scott
Last Updated By	Gavin Scott
Date Created	October 22, 2015
Date Last Updated	October 22, 2015
Actors	Standard CRU User
Description	After signing up to either drive or ride to an event, the user can view information about their ride(s) to and from that event, including the contact information of the driver (if the user is riding) or all the members of the car and the remaining space available (if the user is driving).
Preconditions	The user has already signed up as either a driver or a rider to an event.
Postconditions	None
Normal Flow	<p>1.0 Viewing their Ride Details</p> <ol style="list-style-type: none"> 1. The user navigates to the ride sharing page of the app. 2. The user expands an event to which they already have a ride associated. 3. A list of the user's rides associated with that event are displayed, the icons of which includes the name of the driver and the date and time of the rides. <ol style="list-style-type: none"> a. If the user is a driver, the ride description also includes the number of available seats remaining. 4. After selecting a ride, the system displays the contact information they will need to use to make contact with the members of the car. <ol style="list-style-type: none"> a. If the user is a driver, the contact information page displays the first name and number of everyone that has already enrolled for a ride in their car. b. If the user is a rider, the page only displays the first name and number of the driver.
Alternative Flows	None
Exceptions	None
Priority	High
Special Requirements	None
Assumptions	The user has already signed up to either drive or ride to the selected event.

3.8 Login

Use Case ID:	8
Use Case Name:	Login
Created by:	Mason Stevenson
Last Updated by:	Mason Stevenson
Date Last Updated:	10/17/15
Actors:	Community Group Leader
Description:	Community group leaders can log in to the app to view extra resources.
Preconditions:	1. The Community Group Leader must have an account registered.
Postconditions:	1. The app must now allow the community group leader to view extra resources.
Normal Flow:	1. The user navigates to the leader resources section. 2. The user enters his email and password. 3. The app now allows access to the leader resources section.
Alternative Flows:	1. The user entered an incorrect email/password combination. a. The user is notified that his information is incorrect. b. Return to step 1. 2. The user forgot his password. a. The app asks the user for his email. b. The user receives an email with password reset instructions. c. Return to step 1.
Exceptions:	None
Includes:	None
Priority:	High
Frequency of use:	Very low usage
Business Rules:	None
Special Requirements:	None

3.9 Join a Ministry

Use Case ID	9
Use Case Name	Join a Ministry
Created By	Kayla Carr
Last Updated By	Kayla Carr
Date Created	10/17/2015
Date Last Updated	10/17/2015
Actors	Existing members of Cru Community leaders of Cru Potential new members of Cru
Description	
Preconditions	1. User has downloaded the app
Postconditions	1. User has indicated which campus and ministry they are joining 2. System has recorded which campus and ministry the user is in
Normal Flow	1.0 Join a Ministry 1. User launches app for the first time (Alternative: if not first launch, see 1.1) 2. System displays a list of campuses 3. User selects which campus(es) they are a part of 4. System displays a list of ministries available at each campus the user selected 5. User selects which ministr(ies) they are a part of 6. System records the user's ministries and campuses in the app 7. System begins displaying information related to those ministries and campuses within the Events, Resources, and other tabs
Alternative Flows	1.1 Editing Ministries 1. User navigates to their settings 2. User indicates they would like to change their ministry and/or campus 3. User re-selects their campus(es) 4. User re-selects their ministr(ies) 5. User saves the new settings 6. System updates the user's campus and ministry data
Exceptions	None
Priority	High

Special Requirements	None
----------------------	------

3.10 Change or Add Campuses

Use Case ID:	10
Use Case Name:	Change or Add Campuses
Created By:	Gavin Scott
Last Updated By:	Gavin Scott
Date Created:	October 17, 2015
Date Last Updated:	October 17, 2015
Actors:	Standard CRU User
Description:	Users select campuses when they install the app, but they have the option to modify their list of campuses at any point. They must always be subscribed to at least one campus.
Preconditions:	None
Postconditions:	The user's list of subscribed campuses has been updated.
Normal Flow:	<p>1.0 The user adds campuses</p> <ol style="list-style-type: none"> 1. The user accesses the settings menu and selects the option to change their subscribed campuses. 2. The user is presented with a list of the campuses they are currently subscribed to, each with the option to remove, and an option to add a new campus. 3. The user selects the option to add a new campus 4. The system displays a list of all available, unsubscribed campuses to the user. 5. The user selects as many campuses as they would like and submits the data. 6. The system returns to the list of subscribed campuses, showing the changes and a confirmation message. <p>1.1 The user removes a campus</p> <ol style="list-style-type: none"> 1. Begin on step 1.0.2. 2. The user selects the option to remove a campus. 3. The system removes the campus from the list and displays a confirmation message.
Alternative Flows:	<p>1.2 The user attempts to remove all campuses</p> <ol style="list-style-type: none"> 1. The final campus is not removed, and the system displays an error

	message explaining the requirement that every user be subscribed to at least one campus.
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Low usage per user, most users will never change or add campuses
Business Roles:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.11 View Ministry Team Information

Use Case ID:	11
Use Case Name:	View Ministry Team Information
Created By:	Brian Quezada
Last Updated By:	Brian Quezada
Date Created:	October 18, 2015
Date Last Updated:	October 22, 2015
Actors:	Standard CRU User
Description:	The user can view information about teams in order to get involved with Cru. It will have a way to sign-up or get in contact with someone in order to get more information about joining the team.
Preconditions:	None
Postconditions:	None
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to “Get Involved” 2. User selects a team from a list of Ministry Teams 3. Information is displayed about the team and the people who run the team, along with a link to the team’s facebook group

Alternative Flows:	<ol style="list-style-type: none"> 1. User signs up for information <ol style="list-style-type: none"> 1.1. User goes to “Get Involved” 1.2. User selects a team 1.3. User selects sign up for team 1.4. There is a prompt for information such as phone number and name 1.5. The user may be contacted by Cru personnel independent of the application.
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Low usage
Business Roles:	TBD
Special Requirements:	TBD
Assumptions:	None
Notes and Issues:	None

3.12 View and Apply to a Summer Mission

Use Case ID:	12
Use Case Name:	Join a Summer Mission
Created By:	Brian Quezada
Last Updated By:	Brian Quezada
Date Created:	October 22, 2015
Date Last Updated:	October 22, 2015
Actors:	Standard CRU User
Description:	The user can view information about summer missions in order to get involved with Cru mission trips. Relevant information should be available and a link to an application website should be available.
Preconditions:	None
Postconditions:	None

Normal Flow:	<ol style="list-style-type: none"> 1. User goes to “Summer Missions” 2. User selects a Mission Trip from a list of Mission Trips. 3. The user is presented with information about the trip such as a description, length, cost, leadership, and location. 4. The user select apply and is sent to an outside website to apply for the mission.
Alternative Flows:	<ol style="list-style-type: none"> 1. Find more information about a mission. <ol style="list-style-type: none"> 1.1. User goes to “Summer Missions” 1.2. User select a Mission Trip from a list of Mission Trips. 1.3. The user can access a link to the Mission website for more information.
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Low usage
Business Roles:	TBD
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.13 Join a Ministry Team

Use Case ID	13
Use Case Name	Join a Ministry Team
Created By	Shelli Crispen
Last Updated By	Shelli Crispen
Date Created	October 25, 2015
Date Last Updated	October 25, 2015
Description	A CRU member is trying to learn more about and get involved in a Ministry Team.
Preconditions	None

Natural Flow	1.0 Sign up for a Ministry Team <ol style="list-style-type: none"> Members open the app and Click on the Get Involved tab Then selects the Team they want to get involved with Click the button to “Sign-up” Enter their contact information and click “Ok” Shows a popup of the leaders contacted information with a button “Save to Contacts”. Takes them to the online sign up
Alternative Flow	1.1 Find out more about a Ministry Team <ol style="list-style-type: none"> Members open the app and Click on the Get Involved tab Then selects the Team they want to learn more about Shows information about the team and how to sign up.
Priority	Medium
Frequency of Use:	Low usage
Business Roles:	TBD
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.14 Add an Event to Calendar

Use Case ID	14
Use Case Name	Add Event to Calendar
Created By	Mallika Potter
Last Updated By	Mallika Potter
Date Created	10/25/2015
Date Last Updated	10/25/2015
Actors	Existing members of Cru
Description	Allows the user to add an event to their personal calendar
Preconditions	<ol style="list-style-type: none"> User has joined a ministry or ministries User has joined a campus
Postconditions	1. Event has been added to personal Android Calendar

Normal Flow	1.0 Add Event to Calendar <ol style="list-style-type: none"> 1. User accesses the “events” section of the application. 2. User views events for their ministries.. 3. User selects event. 4. User selects “Save to Calendar.”. 5. Event is added to local calendar.
Alternative Flows	None
Exceptions	None
Includes:	None
Priority	High
Special Requirements	None
Frequency of use:	High usage
Business Rules:	None
Assumptions:	None
Notes and Issues:	None

3.15 Cancel a Ride

Use Case ID	15
Use Case Name	Cancel a Ride
Created By	Shelli Crispen
Last Updated By	Shelli Crispen
Date Created	10/29/2015
Date Last Updated	10/29/2015
Actors	Existing members of Cru
Description	Allows the user to cancel a ride they signed up for.
Preconditions	<ol style="list-style-type: none"> 1. User has joined a ministry or ministries 2. User has joined a campus
Postconditions	<ol style="list-style-type: none"> 1. Ride no longer shows in their personal ride section.

Normal Flow	1.0 Cancel a Ride <ol style="list-style-type: none"> 1. Users go to the “RideShare” tab. 2. Click on “My Rides” 3. Selects the ride they want to cancel from the list of their rides. 4. A popup asks “Would you like to cancel?” 5. User clicks yes and the ride is no longer in their rides list.
Alternative Flows	None
Exceptions	None
Includes:	None
Priority	High
Special Requirements	None
Frequency of use:	High usage
Business Rules:	None
Assumptions:	None
Notes and Issues:	None

4 System Features

4.1 Subscribe to a Ministry

4.1.1 Description and Priority

A member should be able to subscribe to a ministry in order to quickly access that ministry’s resources and receive updates and event information about that ministry.

This a high priority item.

4.1.2 Stimulus/Response Sequences

SR 4.1.1	Stimulus: A member subscribes to a ministry.
	Response: There is some visual feedback that the user has subscribed to the ministry.

4.1.3 Functional Requirements

FR-4.1.1	The system shall prompt the user to enter their campus(es) and interested ministries on initial launch.
FR-4.1.2	The system shall display the list of ministries available only at the user's campus(es).
FR-4.1.3	The system shall allow the user to change which ministries they are involved with.
FR-4.1.4	The system shall display only the current events within the user's involved ministries and campuses.
FR-4.1.5	The system shall allow the user to add events to their Google Calendar.
FR-4.1.6	The system shall allow the user to subscribe/unsubscribe from ministry push notifications.
FR-4.1.7	The system shall remember and automatically detect the user's campuses and ministries.
FR-4.1.8	The system shall allow the user to change which campuses they are involved with.

4.2 Access Ministry Resources

4.2.1 Description and Priority

A user should be able to access articles, videos, and other resources of a ministry. Resources should be relevant to the given ministry. There needs to be an archive of these resources.

This feature has a high priority.

4.2.2 Stimulus/Response Sequences

SR 4.2.1	Stimulus: A user accesses an article from a Cru ministry.
	Response: The article is presented and readable without eye strain. A user can navigate with scrolling up and down. There should be no horizontal movement across the page. Any extra content such as pictures should be displayed in line with the article.
SR 4.2.2	Stimulus: A user selects a video from the Cru ministry

	Response: The application displays the video, ideally in the application. The video loads and automatically plays the desired video.
--	--------------------------------------------------------------------------------------------------------------------------------------

4.2.3 Functional Requirements

FR-2.3.1	The system shall use asynchronous database queries for all queries from the server.
FR-2.3.2	The system shall query the Cru Central Coast server database for content, pictures, and other relevant information about a resource.
FR-2.3.3	The system shall check that its local database is the most recent version.
FR-2.3.4	The system shall query for the most recent database version if its local version is not the most recent.
FR-2.3.5	The system shall display YouTube videos through the YouTube API.
FR-2.3.6	The system shall link to specific playlists on the Cru Central Coast YouTube account.
FR-2.3.7	The system shall open videos in YouTube using implicit intentions.
FR-2.3.8	The system shall display articles from the Cru website.
FR-2.3.9	The system shall open articles within a mobile-friendly in-app interface.
FR-2.3.10	The system shall integrate both text and images when viewing articles.
FR-2.3.11	The system shall display the articles posted by date.
FR-2.3.12	The system shall display videos by date.
FR-2.3.13	The system shall play audio files within articles.
FR-2.3.14	The system shall display links to Cru social media accounts.
FR-2.3.15	The system shall allow the user to search/filter through videos, articles, and audio files.

4.3 Push Notifications

4.3.1 Description and Priority

Allows the user to receive push notifications for ministry events and rides, add ministry events to their Google Calendar, or opt out of further notifications.

High priority item.

4.3.2 Stimulus/Response Sequences

SR 4.3.1	Stimulus: The user tells the system it wants to receive push notifications for events
	Response: The system creates notifications for all of the events in the subscribed ministry
SR 4.3.2	Stimulus: The user tells the system it wants to stop receiving push notifications for events
	Response: The system removes notifications for all of the events in the subscribed ministry
SR 4.3.3	Stimulus: A banked notification is due to occur
	Response: The system sends a notification to the user's phone

4.3.3 Functional Requirements

FR-4.3.1	The system shall allow the user to subscribe to push notifications.
FR-4.3.2	The system shall allow the user to unsubscribe to push notifications.
FR-4.3.3	The system shall send notifications to the user 3 hours before an event is scheduled to occur.
FR-4.3.4	The system shall be able to send notifications at admin-specified times.
FR-4.3.5	The system shall send the user a notification if they are looking for a ride but none were available, then a ride opened up.
FR-4.3.6	The system shall send drivers a notification if their car is filled.

4.4 Access Leader Resources

4.4.1 Description and Priority

In addition to the regular student resources in the resources section, community group leaders will have access to an additional 50-60 articles organized by topic. If the user is logged in to the app as a community leader, the community leader resources section, which is in a subsection separate from student resources but still within the resources section of the app, will become accessible.

High priority item.

4.4.2 Stimulus/Response Sequences

SR 4.4.1	Stimulus: A member accesses the resources section.
	Response: The system displays available member resources.
SR 4.4.2	Stimulus: A community leader accesses the resources section.
	Response: The system displays both available member resources and available community leader resources.
SR 4.4.3	Stimulus: From the resources section, a member logs in.
	Response: The system grants access to the leader resources section.

4.4.3 Functional Requirements

FR-4.4.1	The system shall perform a check to determine if a user is logged in before displaying community leader resources.
FR-4.4.2	The system shall make community leader resources not accessible when a user is logged out.
FR-4.4.3	The system shall make community leader resources accessible when a user is logged in.
FR-4.4.4	The system shall allow for nestable resource categories.

4.5 Join a Community Group

4.5.1 Description and Priority

A Cru member should be able to register for a community group. This involves sending contact information to the community group leader and saving the community group leader's contact information.

This feature is high priority.

4.5.2 Stimulus/Response Sequences

SR-4.5.1	Stimulus: User selects "register for a community group." Response: System displays available times screen.
SR-4.5.2	Stimulus: User selects available times and current year.
	Response: System displays available groups screen.
SR-4.5.3	Stimulus: User selects community group.
	Response: System displays contact information collection screen.
SR-4.5.4	Stimulus: User enters contact information.
	Response: System displays community group leader contact information.
SR-4.5.5	Stimulus: User saves contact information.
	Response: System saves contact information to phone.

4.5.3 Functional Requirements

Requirement	Description
FE-4.5.1	The system shall allow the user to select a class year.
FE-4.5.2	The system shall allow the user to select available times.
FE-4.5.3	The system shall allow the user to input contact information.

FE-4.5.4	The system shall notify the community group leader through text that a new user has joined the community group.
FE-4.5.5	The system shall query the databases for community group leader contact information and display it to the user.
FE-4.5.6	The system shall display ministry specific community group questions, if any.
FE-4.5.7	The system shall display the community leader's name and phone number to the user after they have joined a community group.

4.6 Sign up as a Driver for an Event

4.6.1 Description and Priority

A user should be able to register as a driver for any CRU event that has ride-sharing enabled. This will include the user exchanging their name and phone number with the other members that they will be driving to the event.

This is a high priority feature.

4.6.2 Stimulus/Response Sequences

SR 4.6.1	Stimulus: User selects "I Can Drive!" on a page for an event that has ride-sharing enabled.
	Response: The system displays a splash screen prompting the user for the number of seats they have available.
SR 4.6.2	Stimulus: The user enters an invalid number of seats and presses "Submit"
	Response: The system displays an error message informing the user of the problem.
SR 4.6.3	Stimulus: The user enters a valid number of seats and presses "Submit"
	Response: The System displays a confirmation message
SR 4.6.4	Stimulus: The user presses "OK" on the confirmation message
	Response: The system displays the event page, and the "I Can Drive!" button has been replaced with a "My Car" button.

4.6.3 Functional Requirements

FR 4.6.1	The system shall allow drivers to input their name, cellphone number, time of departure, departure location, and whether they are planning a one-way or round-trip.
FR 4.6.2	The system shall allow drivers to input the number of spaces available in their car.
FR 4.6.3	The system shall specify a default pick-up location when a ride is created.
FR 4.6.4	The system shall allow drivers to specify a pick-up location.
FR 4.6.5	The system shall send a notification to drivers when their car is full.
FR 4.6.6	The system shall verify that drivers are registered members of Cru via their phone number.
FR 4.6.7	The system shall only allow verified members to create a ride.
FR 4.6.8	The system shall show a pop-up error message notifying a user that they are not verified to create a ride and should contact Cru if they have not been verified.
FR 4.6.9	The system shall only allow users to sign up as drivers for events that have ride-sharing enabled.
FR 4.6.10	The system shall remove all rides once an event is no longer available for ride-sharing.
FR 4.6.11	The system shall allow drivers to cancel a ride.
FR 4.6.12	The system shall allow drivers to view their rides.
FR 4.6.13	The system shall allow drivers to view the names and numbers of the people in their ride(s).
FR 4.6.14	The system shall count how many people are in each car.

4.7 Sign up for a Ride to an Event

4.7.1 Description and Priority

A member of Cru should be able to find a ride to an event from other people using the application.

This is a high priority feature.

4.7.2 Stimulus/Response Sequences

SR 4.7.1	Stimulus: User selects the “RideShare” button on a page for an event that has ride-sharing enabled.
	Response: The system displays a screen prompting the user to input a time to leave and then select from a number of options for a ride.
SR 4.7.2	Stimulus: User selects “Get a ride” from the different choices on the screen.
	Response: The system displays a splash screen asking them to enter their phone number.

4.7.3 Functional Requirements

FR 4.7.1	The system shall notify riders if their ride has been cancelled.
FR 4.7.2	The system shall allow riders to enter in their phone number, name, and whether they want a one-way or two-way trip.
FR 4.7.3	The system shall tell riders if there are no open rides for an event, then notify them when a ride is opened.
FR 4.7.4	The system shall allow riders to add themselves to a car pool.
FR 4.7.5	The system shall allow riders to remove themselves from a car pool.
FR 4.7.6	The system shall display a list of all available (not full) rides to the event.
FR 4.7.7	The system shall order rides in descending order based on how close the ride departure time is to the user’s intended departure time.
FR 4.7.8	The system shall allow riders to view their rides.
FR 4.7.9	The system shall allow riders to view their driver(s) name and phone number.
FR 4.7.10	The system shall allow riders to cancel their rides.

4.8 Access Ministry Events

4.8.1 Description and Priority

A member of Cru should be able to view information about ministry events and join the events to get relevant notifications.

This is a high priority feature.

4.8.2 Stimulus/Response Sequences

SR 4.8.1	Stimulus: User selects the Events tab.
	Response: The system displays a list of events from ministries and campuses the user is subscribed to.
SR 4.8.2	Stimulus: User selects an event from the event list
	Response: Date, destination, and other relevant information are displayed for the trip.
SR 4.8.3	Stimulus: User adds the event to their calendar.
	Response: A calendar event is created in the phone's native calendar application or in Google Calendar.
SR 4.8.4	Stimulus: User selects ride sharing from an event page.
	Response: The user is sent to the ride sharing page for the selected event.

4.8.3 Functional Requirements

FR-2.8.1	The system shall use asynchronous database queries for all queries from the server.
FR-2.8.2	The system shall query the Cru Central Coast server database for locations, trip ID's, and date updated fields.
FR-2.8.3	The system shall check that its local database is the most recent version.
FR-2.8.4	The system shall query for the most recent database version if its local version is not the most recent.
FR-2.8.5	The system shall pull current events from the Cru database.
FR-2.8.6	The system shall display current events within an in-app interface.
FR-2.8.7	The system shall allow the user to add events to their Google Calendar.
FR-2.8.8	The system shall update Google Calendar created events if the event in the Cru database changes.

4.9 Join a Summer Mission

4.9.1 Description and Priority

A user should be able to view a list of upcoming summer missions for their campus and/or ministries, view information about them, and sign up for them.

This is a medium priority feature.

4.9.2 Stimulus/Response Sequences

SR 4.9.1	Stimulus: User selects the summer missions section of the app.
	Response: System displays a list of summer missions available to the user.
SR 4.9.2	Stimulus: User selects a summer mission
	Response: System displays information about the summer mission.
SR 4.9.3	Stimulus: User signs up for the summer mission.
	Response: System redirects the user to a webpage where they can join the summer mission.

4.9.3 Functional Requirements

FR-2.9.1	The system shall display summer missions in order of upcoming date.
FR 2.9.2	The system shall only display summer missions that are available for the user's campus and/or ministries.

5 External Interface Requirements

5.1 User Interfaces

UI Requirement	Description
UI-1	The user interfaces shall comply with Cru design standards.
UI-2	The user interfaces shall match current Cru marketing platforms, including colors and logos.

UI-3	The user interfaces shall adhere to the human interface guidelines for Android.
------	---------------------------------------------------------------------------------

5.2 Hardware Interfaces

HI Requirement	Description
HI-1	The application shall run on Android devices.

5.3 Software Interfaces

SI Requirement	Description
SI-1	The application shall pull data from Cru databases.

5.4 Communications Interfaces

CI Requirement	Description
CI-1	The system will not transmit any metrics data to the databases of any kind.
CI-2	The system will request information from the CRU databases using the internet and standard database-querying APIs.

6 Other Nonfunctional Requirements

6.1 Performance Requirements

PR-1	The system shall load information from the Cru database within 10 seconds.
PR-2	The system shall only be down when Cru's database or website are down or inaccessible.
PR-3	The system shall be able to handle a few hundred concurrent users, particularly having a number of users use the ride-sharing feature.

PR-4	The system shall update when a car is full within 10 seconds so members don't accidentally sign up for a full car.
PR-5	The system shall notify the user of any errors (connectivity, etc.) and make it simple to correct them or retry the action later.

6.2 Safety Requirements

SaR-1	The system shall not provide a means for anyone to obtain information about any user that could allow them to locate or harm the user, either physically or otherwise.
SaR-2	The system shall provide some method of verifying the identities of members offering rides to preserve a measure of safety for the ride receiver.

6.3 Security Requirements

SR-1	The system shall not store user passwords on the device to prevent passwords from being datamined.
SR-2	The system shall not let community group members' names and contact information be viewable by anyone but log-in verified users.
SR-3	The login system shall be secured with a TLS protocol.
SR-4	User info, such as name and number, should not be broadcast to any other members unless the user chooses to broadcast it.

6.4 Software Quality Attributes

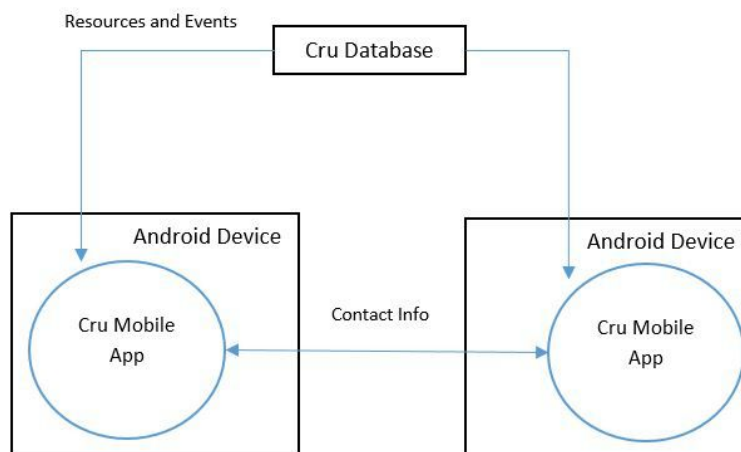
SQA-1	The system shall be modularized.
SQA-2	The system shall be easily interchangeable with differing database systems.
SQA-3	The system shall follow the Cru Coding Style Guide.
SQA-4	The system shall be well documented.
SQA-5	Community leader resources should support easy introduction of new content.
SQA-6	The system shall work properly on all versions of Android after and including Jellybean.

A Glossary

Term	Definition
Ministry	A sub-organization of Cru usually affiliated with a single campus.
Ministry Team	Specialized groups that perform specific tasks to aid a ministry or event.
Community Group	A smaller group inside a ministry.
Community Group Leader	An individual who organizes a community group.
Event	Recurring and once only meetings and activities related to one or more ministries.
Summer Mission	Special events that last several weeks and involve travel.

B Analysis Models

B.1 Context Diagram



C Issues List

There are no issues at this time.