

Cru iOS App: Software Requirements Specification version 3.0

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

December 2, 2015

Contents

Credits	4
1 Introduction	6
1.1 Purpose	6
1.2 Document Conventions	6
1.3 Intended Audience and Reading Suggestions	6
1.4 Project Scope	6
1.5 References	6
2 Overall Description	7
2.1 Product Perspective	7
2.2 Product Features	7
2.3 User Classes and Characteristics	7
2.4 Operating Environment	8
2.5 Design and Implementation Constraints	8
2.6 User Documentation	8
2.7 Assumptions and Dependencies	8
2.8 Business Rules	9
3 Use Cases	9
3.1 Use Case 1: Join a Community Group	9
3.2 Use Case 2: Join a Ministry Team	11
3.3 Use Case 3: See Joined Community Groups and Ministry Teams	13
3.4 Use Case 4: Apply to a Summer Missions Trip	14
3.5 Use Case 5: Offer a Ride	15
3.6 Use Case 6: Request a Ride	18
3.7 Use Case 7: Cancel a Ride	20
3.8 Use Case 8: View the Latest Meeting Recap	22
3.9 Use Case 9: Sync Calendar to iOS / Google Calendar	23
3.10 Use Case 10: Looking for Summer Missions Information	24
3.11 Use Case 11: View Hidden Resources	25
3.12 Use Case 12: Selecting multiple ministries	27
3.13 Use Case 13: Viewing Events	28
3.14 Use Case 14: Viewing Ministries	29
3.15 Use Case 15: Viewing Ride Status As A Driver	30
3.16 Use Case 16: Viewing Ride Status As A Passenger	31

3.17	Use Case 17: Receiving Notification of an Event	32
3.18	Use Case 18: Viewing Guides	33
3.19	Use Case 19: Viewing Articles and Studies	34
3.20	Use Case 20: Viewing Other Sites	35
3.21	Use Case 21: Syncing Academic Calendar	36
3.22	Use Case 22: Unsyncing Academic Calendar	37
3.23	Use Case 23: Viewing Campus Map	38
3.24	Use Case 24: Choosing Initial Campus and Ministry	39
4	System Features	40
4.1	Ride Sharing	40
4.1.1	Description and Priority	40
4.1.2	Stimulus/Response Sequences	41
4.1.3	Functional Requirements	41
4.2	Event Subscriptions	41
4.2.1	Description and Priority	41
4.2.2	Stimulus/Response Sequences	42
4.2.3	Functional Requirements	42
4.3	Cru Resources	42
4.3.1	Description and Priority	42
4.3.2	Stimulus/Response Sequences	42
4.3.3	Functional Requirements	43
4.4	Get Involved	43
4.4.1	Description and Priority	43
4.4.2	Stimulus/Response Sequences	43
4.4.3	Functional Requirements	44
4.5	Summer Missions	44
4.5.1	Description and Priority	44
4.5.2	Stimulus/Response Sequences	44
4.5.3	Functional Requirements	44
4.6	My School	45
4.6.1	Description and Priority	45
4.6.2	Stimulus/Response Sequences	45
4.6.3	Functional Requirements	45
5	External Interface Requirements	45
5.1	User Interfaces	45
5.2	Hardware Interfaces	46

5.3	Software Interfaces	46
5.4	Communications Interfaces	46
6	Other Nonfunctional Requirements	47
6.1	Performance Requirements	47
6.2	Safety Requirements	47
6.3	Security Requirements	47
6.4	Software Quality Attributes	47

Credits

Name	Date	Role	Version
Daniel Lee	October 28, 2015	Lead Author of Introduction and Use Cases	2.0
Jordan Tang	October 28, 2015	Lead Author of Overall Description and Use Cases	2.0
Mariel Sanchez	October 28, 2015	Lead Author of System Features and Use Cases	2.0
Eric Tran	October 28, 2015	Lead Author of External Interface Requirements and Use Cases	2.0
Tammy Kong	October 28, 2015	Lead Author of Other Nonfunctional Requirements and Use Cases	2.0

Revisions

Name	Date	Role	Version
Daniel Lee	November 21, 2015	Adding More Use Cases	3.0
Jordan Tang	November 21, 2015	Updating Overall Description and Use Cases	3.0
Mariel Sanchez	November 21, 2015	Updating Functional Requirements	3.0
Eric Tran	November 21, 2015	Adding More Use Cases And Minor Updates to External Interface Requirements	3.0
Tammy Kong	November 21, 2015	Updating Other Nonfunctional Requirements And Adding More Use Cases	3.0

1 Introduction

1.1 Purpose

The product we are building for Cru Central Coast is an iOS mobile application with the purpose of connecting students to the organization through various efforts, as mentioned in section 2.2 of the Vision and Scope Document. We hope with the 1.0 release that our application will provide ride-sharing capabilities and an intuitive GUI with the integration of Cru's database for login and resource data.

1.2 Document Conventions

This document was written with the general-to-specific pattern, in which broader requirements were broken down into more specific requirements. As a result, priorities for higher-level requirements are assumed to be inherited by detailed requirements.

1.3 Intended Audience and Reading Suggestions

Intended readers for this document include developers, project managers, documentation writers, testers, and release managers. The document is organized and ordered by the process our Software Engineers followed in understanding Cru's proposal. Therefore, all readers except testers are expected to start from the introduction and incrementally proceed through each section for understanding the entire product. Testers may begin reading from Section 3 and progress through all of the requirements to create and employ a test plan for the product.

1.4 Project Scope

See sections 3.1 and 3.2 of the Vision and Scope Document.

1.5 References

- Cru Proposal by Kyle Fletcher
- Cru Brand Identity Guidelines, May 2012

- Vision and Scope Document v1.0 by iCrew
- Cru's Java Style Guide prepared by Kyle Fletcher
- Cru's Swift Style Guide prepared by Kyle Fletcher
- <http://www.crucentralcoast.com/>
- <https://instagram.com/slocru/>
- <https://www.youtube.com/user/slocrusade/>

2 Overall Description

2.1 Product Perspective

The app being created for the Cru organization is a new, self-contained product. The idea for the app originated from the need to ease communication between the expanding Cru organization. The app will be available for all users, from new Cru students to people wanting to ride share with each other.

2.2 Product Features

See section 2.2 in Vision and Scope Document.

2.3 User Classes and Characteristics

User Class	Description
Student Cru members	These are the standard users of the app. They will have access to a smartphone and they'll want to find rides to events, summer missions, or join teams.
Community Group leaders	These users will have access to a smartphone. They'll put in their contact information as well as meeting times so they can find interested members. They will may also want to login to resources in order to get access to leadership guides.

New students	These users are looking to join a Cru ministry or just Cru in general. They will want to find information regarding Cru so they can get a better understanding what the organization is about.
Ride Sharers	These users want to offer rides to other members to Cru events. They will put in their ride and contact location for members to reach out to them.

2.4 Operating Environment

The operating platform for the application will be iOS devices. It will run on the latest version of iOS, iOS 9, to account for the release date in June. We will be programming in Swift. If there is time near the end, we may try to implement a version to work on tablets, but that is not within the current scope of the project.

2.5 Design and Implementation Constraints

The application that we deliver must be maintainable by the Cru organization. This means having clean, well-documented code that can be very easy for someone to look at and pick up.

The application also must be flexible to expansion. It will be created with a list of schools and community groups, but those lists should be able to be easily added to or updated WITHOUT the app having to go through an update through the Play Store.

The application must be modularized. Things like event sharing, ride sharing, or community groups should be able to easily be detached from the app as a separate module, and in turn be easy to plug into another application.

2.6 User Documentation

The bulk of our documentation will be following the comment guidelines on the style sheet. We will be considering creating a wiki to serve as a manual for documentation, depending on effectiveness of our comments.

2.7 Assumptions and Dependencies

See section 2.3 in Vision and Scope Document

2.8 Business Rules

The code for the app must follow the coding style as shown in the coding style guide. A driver offering to give rides must be a registered Cru member under the Cru databases. In order to be a community group leader, the leader must share their contact information to be viewed in the app.

3 Use Cases

3.1 Use Case 1: Join a Community Group

Use Case ID:	1
Use Case Name:	Join a Community Group
Created By:	Daniel Lee
Last Updated By:	Daniel Lee
Date Created:	October 12, 2015
Date Last Updated:	October 12, 2015
Actors:	Prospective Cru Central Coast Member
Description:	A user is interested in joining Cru Central Coast and wishes to get involved. After opening the app, the user navigates to the “Get Involved” section and fills out a questionnaire for getting plugged into a community group. The user then receives contact information of the community group leader.
Preconditions:	<ol style="list-style-type: none">1. The app is already installed on user’s phone2. User correctly fills out questionnaire with appropriate information
Postconditions:	<ol style="list-style-type: none">1. User receives contact information of a community group leader based on his/her preferences
Normal Flow:	1.0 Join a Community Group

	<ol style="list-style-type: none"> 1. User opens app and navigates to “Get Involved” tab 2. System displays ways of getting involved with Cru Central Coast 3. User chooses option to get plugged into a community group 4. System displays questionnaire inquiring for name, phone number, gender, time slot availability and ministry-specific questions 5. User fills out all fields and submits 6. System displays contact information of at least one community group leader with a message to indicate their information being sent to the leader 7. User confirms of the message and returns to “Get Involved” section
Alternative Flows:	<p>1.1 User decides not to enter in personal information (branch at step 5)</p> <ol style="list-style-type: none"> 1. User cancels out 2. Return to step 1
Exceptions:	<p>1.0.E.1 Failure to establish connection to server</p> <ol style="list-style-type: none"> 1. System displays modal “Cannot connect to server” with “Try Again” option 2. User presses “Try Again” until connection is established or “Cancel” to return to main “Get Involved” screen
Includes:	None
Priority:	High
Frequency of Use:	Moderate usage with at least 1 request per user
Business Rules:	None
Special Requirements:	None
Assumptions:	User is not plugged into a Community Group and there exists a community group that fits the user’s schedule
Notes and Issues:	<ol style="list-style-type: none"> 1. If user accidentally enters in incorrect information, should user start over?

3.2 Use Case 2: Join a Ministry Team

Use Case ID:	2
Use Case Name:	Join a Ministry Team
Created By:	Daniel Lee
Last Updated By:	Daniel Lee
Date Created:	October 26, 2015
Date Last Updated:	October 26, 2015
Actors:	Prospective Cru Central Coast Member
Description:	A user is interested in joining Cru Central Coast and wishes to serve or get involved. After opening the app, the user navigates to the “Get Involved” section and fills out a questionnaire for getting plugged into a ministry team. The user then receives contact information of the ministry team leader.
Preconditions:	<ol style="list-style-type: none">1. The app is already installed on user’s phone2. User correctly fills out questionnaire with appropriate information
Postconditions:	<ol style="list-style-type: none">1. User receives contact information of a ministry team leader
Normal Flow:	1.0 Join a Ministry Team

	<ol style="list-style-type: none"> 1. User opens app and navigates to “Get Involved” tab 2. System displays ways of getting involved with Cru Central Coast 3. User chooses option to get plugged into a ministry team 4. System displays list of available ministry teams in user’s campus(es) 5. User chooses one ministry team 6. System displays questionnaire inquiring for name and phone number 7. User fills out both fields and submits 8. System displays contact information of the ministry team leader with a message to indicate their information being sent to the leader 9. User confirms of the message and returns to “Get Involved” section
Alternative Flows:	<p>1.1 User decides not to enter in personal information (branch at step 5)</p> <ol style="list-style-type: none"> 1. User cancels out 2. Return to step 1
Exceptions:	<p>1.0.E.1 Failure to establish connection to server</p> <ol style="list-style-type: none"> 1. System displays modal “Cannot connect to server” with “Try Again” option 2. User presses “Try Again” until connection is established or “Cancel” to return to main “Get Involved” screen
Includes:	None
Priority:	High
Frequency of Use:	Moderate usage with at least 1 request per user
Business Rules:	None
Special Requirements:	None
Assumptions:	User is not plugged into a ministry team
Notes and Issues:	

3.3 Use Case 3: See Joined Community Groups and Ministry Teams

Use Case ID:	3
Use Case Name:	See Joined Community Groups and Ministry Teams
Created By:	Daniel Lee
Last Updated By:	Daniel Lee
Date Created:	October 26, 2015
Date Last Updated:	October 26, 2015
Actors:	Cru Central Coast Member
Description:	A user has already joined a community group and a ministry team but forgot to save the leader's information. After opening the app, the user navigates to the "Get Involved" section and then navigates to currently involved tab. The user then sees a list of community groups and/or ministry teams he/she is involved in and may see their contact information.
Preconditions:	1. The app is already installed on user's phone
Postconditions:	1. User sees joined community groups and ministry teams with each of their contact information
Normal Flow:	1.0 See Joined Community Groups and Ministry Teams 1. User opens app and navigates to "Get Involved" tab 2. System displays ways of getting involved with Cru Central Coast and currently involved tab 3. User navigates to currently involved tab 4. System displays list of joined community groups and ministry teams with each of their contact information
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	High
Frequency of Use:	Moderate usage with at least 1 request per user
Business Rules:	None

Special Requirements:	None
Assumptions:	User is plugged into at least one community group and one ministry team
Notes and Issues:	None

3.4 Use Case 4: Apply to a Summer Missions Trip

Use Case ID:	4
Use Case Name:	Apply to a Summer Missions Trip
Created By:	Daniel Lee
Last Updated By:	Daniel Lee
Date Created:	October 26, 2015
Date Last Updated:	October 26, 2015
Actors:	Cru Central Coast Member
Description:	A user looking to get more involved with Cru has decided to apply for a summer missions trip. After opening the app, the user navigates to the “Summer Missions” section and sees a list of available summer missions opportunities. The user then chooses one and hits apply which takes the user to an external web browser with the application form.
Preconditions:	<ol style="list-style-type: none"> 1. The app is already installed on user’s phone 2. User correctly fills out questionnaire with appropriate information
Postconditions:	<ol style="list-style-type: none"> 1. User sees available summer missions opportunities and can apply to one or more
Normal Flow:	1.0 Apply to a Summer Missions Trip

	<ol style="list-style-type: none"> 1. User opens app and navigates to “Summer Missions” section 2. System displays available summer missions opportunities, each with a description, image, date, location, leaders, and an application method 3. User applies to one summer missions opportunity 4. System opens an external web browser to display application form
Alternative Flows:	None
Exceptions:	1.0.E.1 Failure to establish connection to server <ol style="list-style-type: none"> 1. System displays modal “Cannot connect to server” with “Try Again” option
Includes:	None
Priority:	Medium
Frequency of Use:	Moderate usage with at least 1 request per user
Business Rules:	None
Special Requirements:	None
Assumptions:	There exist summer missions opportunities
Notes and Issues:	

3.5 Use Case 5: Offer a Ride

Use Case ID:	5
Use Case Name:	Offer a Ride
Created By:	Mariel Sanchez
Last Updated By:	Mariel Sanchez
Date Created:	October 12, 2015
Date Last Updated:	October 27, 2015
Actors:	One Cru Central Coast Member

Description:	Users of the app will go to the Events tab to view upcoming events that they have subscribed to. If the event is large or far enough as determined by the event administrator(s), the RideShare option may be available for its users. If RideShare is available, the user will be able to offer a ride both or one way(s) to the event and will receive a notification every time another user is assigned to his/her car.
Preconditions:	<ol style="list-style-type: none"> 1. The app is already installed on user's phone 2. The driver is a registered members of Cru
Postconditions:	<ol style="list-style-type: none"> 1. The driver gets notified that one less seat is available in his/her car 2. The driver receives contact information of the passenger and vice versa
Normal Flow:	1.0 Offer a Ride to an Event

	<ol style="list-style-type: none"> 1. Driver opens the app and selects the events tab 2. System displays the list of upcoming events available for the drivers ministry 3. Driver selects an event that he/she wishes to attend 4. The “RideShare” button is enabled for this event and the Driver selects this button 5. System displays option to offer a ride and request a ride 6. Driver offers a ride 7. System displays questionnaire asking the driver what time he/she wants to leave for the event, the number of seats available in their car, the location for pick up, and whether they can offer the ride both or one way(s) to the event 8. Driver selects both ways and fills out other information in the questionnaire and that offer is now pending 9. System pairs the driver and a passenger together and driver gets notification that one less seat is available in his car 10. Driver receives contact information of the passenger
Alternative Flows:	<ol style="list-style-type: none"> 1.1 Driver offers a ride through RideShare section(continue from step 1) <ol style="list-style-type: none"> 1. Driver selects Offer a Ride 2. System displays questionnaire asking the driver what event, what time he/she wants to leave for the event, and the number of seats available in their car 3. Start again at step 8 1.2 Ride Share is Not Enabled (branch off step 3) <ol style="list-style-type: none"> 1. Ride share option is disabled and Driver is unable to press the button
Alternative Flows:	<ol style="list-style-type: none"> 1.3 Driver can only offer one way to the event (branch off step 8)

	<ol style="list-style-type: none"> 1. Driver selects one way (indicating to or from) and fills out other information in the questionnaire and that offer is now pending 2. Start again at step 9
Exceptions:	<p>1.0.E.1 User is not a registered Cru member (branch from step 6)</p> <ol style="list-style-type: none"> 1. System informs user they are not a registered Cru member and cannot offer a ride
Includes:	None
Priority:	Medium
Frequency of Use:	Over a thousand members of Cru, but only when an event enables this option
Business Rules:	None
Special Requirements:	<ol style="list-style-type: none"> 1. Connection with Cru database to check for members 2. Users who are not in Cru database should still be able to request a ride
Assumptions:	Assume at least 15% of users will offer rides
Notes and Issues:	None

3.6 Use Case 6: Request a Ride

Use Case ID:	6
Use Case Name:	Request a Ride
Created By:	Mariel Sanchez
Last Updated By:	Mariel Sanchez
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One Cru Central Coast Member

Description:	Users of the app will go to the Events tab to view upcoming events that they have subscribed to. If the event is large or far enough as determined by the event administrator(s), the RideShare option may be available for its users. If the RideShare is available, the user will be able to request a ride one or both way(s) to the event and will receive a notification about who his/her driver is before the start of the event.
Preconditions:	1. The app is already installed on user's phone
Postconditions:	<ol style="list-style-type: none"> 1. The passenger gets notified that his/her request has been accepted by a driver with a reminder of meeting time and place 2. The passenger receives contact information of the driver
Normal Flow:	<p>1.0 Request a Ride to an Event</p> <ol style="list-style-type: none"> 1. User opens the app and selects the events tab 2. System displays the list of upcoming events available for the drivers ministry 3. User selects an event that he/she wishes to attend 4. The RideShare button is enabled for this event and the Driver selects this button 5. System displays option to request a ride and offer a ride 6. User requests a ride 7. System displays questionnaire asking the user what time he/she wants to leave for the event, pick up location, and whether he/she needs a ride one or both way(s) to the event 8. User selects both ways and fills out and submits all other information and that request is now pending 9. System pairs the user with a driver user gets a notification that the request gets fulfilled 10. Passenger get contact information of the driver
Alternative Flows:	1.1 User requests a ride through the Rideshare section(continue from step 1)

	<ol style="list-style-type: none"> 1. User requests a ride 2. System displays questionnaire asking the user what event, what time he/she wants to leave for the event, and the number of seats available in their car 3. Start again at step 8 <p>1.2 Ride Share is Not Enabled (branch off step 3)</p> <ol style="list-style-type: none"> 1. Ride share option is disabled and Driver is unable to press the button <p>1.3 User only needs ride one way (branch off of step 8)</p> <ol style="list-style-type: none"> 1. User selects one way, indicating to or from, and fills out and submits all other information and that request is now pending 2. Start again at step 9
Exceptions:	<p>1.0.E.1 User is not a registered Cru member (branch from step 6)</p> <ol style="list-style-type: none"> 1. System informs user they are not a registered Cru member and cannot offer a ride
Includes:	None
Priority:	Medium
Frequency of Use:	Over a thousand members of Cru, so expect most to want to request a ride
Business Rules:	None
Special Requirements:	<ol style="list-style-type: none"> 1. Connection with Cru database to check for members 2. Users who are not in Cru database should still be able to request a ride
Assumptions:	Assume majority of users will make use of this feature
Notes and Issues:	None

3.7 Use Case 7: Cancel a Ride

Use Case ID:	7
Use Case Name:	Cancel a Ride
Created By:	Mariel Sanchez

Last Updated By:	Mariel Sanchez
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	Two Cru Central Coast Member - Passenger and Driver
Description:	Passenger has requested a ride and can no longer attend. The user wishes to cancel the ride.
Preconditions:	<ol style="list-style-type: none"> 1. The app is already installed on the user's phones 2. The passenger has requested a ride and has been assigned to a driver.
Postconditions:	<ol style="list-style-type: none"> 1. The driver gets notified that one more seat in his car is available again
Normal Flow:	<p>1.0 Cancel a Ride to an event</p> <ol style="list-style-type: none"> 1. Passenger opens the app and selects the Rideshare section to view status of his/her event rides 2. System displays the status of the passenger's ride requests and offers 3. Passenger selects an event that he/she requested a ride 4. System displays prompt asking what action to take regarding that event 5. Passenger selects cancelling the request 6. Event is removed from status list 7. Driver gets notified that a passenger has cancelled a ride request
Alternative Flows:	<p>1.1 Driver cancels his/her offer (branch from step 2)</p> <ol style="list-style-type: none"> 1. Passenger gets notified that ride has been cancelled 2. System automatically creates a new request for the passenger and the status is back to pending
Exceptions:	<p>1.0.E.1 System is unable to receive ride status information</p> <ol style="list-style-type: none"> 1. User gets notified of error and needs to restart the application
Includes:	None
Priority:	Medium

Frequency of Use:	All users will be able to cancel an event
Business Rules:	None
Special Requirements:	Connection with Cru database to list ride status
Assumptions:	Assume majority of users will make use of this feature
Notes and Issues:	<ol style="list-style-type: none"> 1. What is driver and passenger cancel at the same time? 2. How will they get notified of cancellation?

3.8 Use Case 8: View the Latest Meeting Recap

Use Case ID:	8
Use Case Name:	View the Latest Meeting Recap
Created By:	Eric Tran
Last Updated By:	Eric Tran
Date Created:	October 13, 2015
Date Last Updated:	October 13, 2015
Actors:	One Student already involved with Cru
Description:	Users of the app will go to the Resources tab to view an archive of all videos from Cru Central Coast's YouTube account. This includes weekly recaps of meetings and other miscellaneous videos. Videos will be shown with most recent videos first by default. Users can choose to filter videos when looking for a specific video.
Preconditions:	<ol style="list-style-type: none"> 1. The app is already installed on student's iPhone 2. The user has internet connection (WiFi/4G)
Postconditions:	<ol style="list-style-type: none"> 1. The student is able to view YouTube video.
Normal Flow:	1.0 Viewing a Cru video

	<ol style="list-style-type: none"> 1. Student opens the app and selects the Resources tab 2. System displays the list of recent videos from Cru Central Coast's YouTube archive 3. User can choose to filter search results. 4. User selects video he/she would like to see 5. Video displays on phone.
Alternative Flows:None	
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Once a week
Business Rules:	None
Special Requirements:	<ol style="list-style-type: none"> 1. Connection to Cru Central Coast's YouTube account 2. Internet connectivity (WiFi/4G)
Assumptions:	There will be at multiple videos to display from the archive
Notes and Issues:	None

3.9 Use Case 9: Sync Calendar to iOS / Google Calendar

Use Case ID:	9
Use Case Name:	Sync Calendar to iOS / Google Calendar
Created By:	Jordan Tang
Last Updated By:	Jordan Tang
Date Created:	October 13, 2015
Date Last Updated:	October 13, 2015
Actors:	One Student already involved with Cru

Description:	Users of the app will want to sync the event calendar with their own iOS / Google calendar in order to have personal notifications for events. The user will navigate to the “Events” tab. From there, they will be able to sync the calendar with their corresponding device. Once selected, the user will put in their necessary credentials in order to properly sync.
Preconditions:	<ol style="list-style-type: none"> 1. The app is installed on the user’s phone 2. The user has an valid Google / Apple account 3. The user has Wi-Fi / 4G connection
Postconditions:	<ol style="list-style-type: none"> 1. Event information appears on student’s calendar
Normal Flow:	<p>1.0 Syncing Calendar</p> <ol style="list-style-type: none"> 1. User opens up the Cru app 2. User navigates to Events tab 3. User notifies to sync in the events tab 4. User selects which calendar to sync to 5. User enters necessary credentials 6. User confirms and should see updated calendar
Alternative Flows:	None
Exceptions:	<p>1.0.E.1 User does not have an iOS / Google account</p> <ol style="list-style-type: none"> 1. Cru cannot sync without account credentials
Includes:	None
Priority:	Medium
Frequency of Use:	Maybe once a couple of weeks to account for new events
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.10 Use Case 10: Looking for Summer Missions Information

Use Case ID:	10
--------------	----

Use Case Name:	Looking for Summer Missions Information
Created By:	Tammy Kong
Last Updated By:	Tammy Kong
Date Created:	October 13, 2015
Date Last Updated:	October 13, 2015
Actors:	One Student already involved with Cru
Description:	User will be able to navigate through the iOS application to find a list of Cru summer missions as well as specific details about each mission such as description, mission leader, duration, and location.
Preconditions:	<ol style="list-style-type: none"> 1. The app is installed on the user's phone 2. The user has Wi-Fi / 4G connection
Postconditions:	<ol style="list-style-type: none"> 1. User will be able to view a list of summer missions offered by Cru
Normal Flow:	1.0 Scrolling through Summer Missions <ol style="list-style-type: none"> 1. User opens up the Cru app 2. User navigates to "Summer Missions" section 3. User scrolls through the list of Summer Missions
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	Low
Frequency of Use:	Maybe once a couple of weeks (as it gets closer to the summer) to account for new summer missions
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.11 Use Case 11: View Hidden Resources

Use Case ID:	11
Use Case Name:	View Hidden Resources
Created By:	Tammy Kong

Last Updated By:	Tammy Kong
Date Created:	October 13, 2015
Date Last Updated:	October 13, 2015
Actors:	One Community Leader from Cru
Description:	The user wants to check if there is any newly updated resources for the community leaders.
Preconditions:	<ol style="list-style-type: none"> 1. The app is installed on the user's phone 2. The user has Wi-Fi / 4G connection 3. The user is a community leader
Postconditions:	<ol style="list-style-type: none"> 1. The user is able to view the hidden resources that are only viewable by community leaders.
Normal Flow:	<p>1.0 Viewing hidden resource</p> <ol style="list-style-type: none"> 1. User opens up the Cru app 2. User selects the "Resource" section 3. User logs in 4. User inputs username and password 5. "Resource" section will refresh and hidden resources will be viewable
Alternative Flows:	<p>1.0 User logs out (continuing from step 5)</p> <ol style="list-style-type: none"> 1. User logs out 2. Resource section will be refreshed and resources for the community leaders are then hidden
Exceptions:	<p>1.0.E.1 Failure to establish connection to server</p> <ol style="list-style-type: none"> 1. System notifies user connection was not established and will prompt user to try again. 2. Start again on step 3 <p>1.0.E.2 Member inputs invalid information (branch from step 4)</p> <ol style="list-style-type: none"> 1. System notifies user of invalid information and will prompt user to try again. 2. Start again on step 3 <p>1.0.E.3 User is not a community leader</p>

	1. User does not have a username and password information
Includes:	None
Priority:	High
Frequency of Use:	Low usage of about once a week per community leader
Business Rules:	None
Special Requirements:	User is a community leader and has a username and password to use for login.
Assumptions:	None
Notes and Issues:	None

3.12 Use Case 12: Selecting multiple ministries

Use Case ID:	12
Use Case Name:	Selecting multiple ministries
Created By:	Eric Tran
Last Updated By:	Eric Tran
Date Created:	October 25, 2015
Date Last Updated:	October 25, 2015
Actors:	One student already involved with Cru
Description:	A user wishes to get involved with multiple specific ministries. The user opens the app navigates to the “Get Involved” tab. Within this section, the user will select ministries they are interested in.
Preconditions:	1. The app is already installed on student’s iPhone
Postconditions:	1. Student’s app saved the ministries student is interested in.
Normal Flow:	1.0 Selecting multiple ministries 1. Student opens the app 2. Student goes to “Get Involved” 3. User proceeds to select the multiple ministries they are interested with. 4. Application saves user preferences.
Alternative Flows:	1.0 Selecting multiple ministries during initial launch

	<ol style="list-style-type: none"> 1. Student opens the app 2. App displays campuses for student to select 3. Student selects campuses they are associated with 4. App displays ministries for student to select 5. Application saves user preferences
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Once during initial setup/Rarely, only when ministry interests change
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.13 Use Case 13: Viewing Events

Use Case ID:	13
Use Case Name:	Viewing Events
Created By:	Jordan Tang
Last Updated By:	Jordan Tang
Date Created:	October 25, 2015
Date Last Updated:	October 25, 2015
Actors:	One student interested in Cru/already involved with Cru
Description:	A user opens the app and wishes to view the current possible events to attend within their associated campuses and ministries.
Preconditions:	<ol style="list-style-type: none"> 1. The app is already installed on student's iPhone 2. The user has WiFi / 4G connectivity
Postconditions:	<ol style="list-style-type: none"> 1. Student has knowledge of upcoming Cru Central Coast Events
Normal Flow:	1.0 Viewing Events

	<ol style="list-style-type: none"> 1. Student opens the app 2. App automatically opens to the event page (default) 3. A scrollable list of events will be viewable by the user, sorted by nearest date
Alternative Flows:	1.0 Viewing Events from Another Selection <ol style="list-style-type: none"> 1. Student already has the app open on another tab 2. Student will press events tab on the bottom 3. App will open the events page 4. A scrollable list of events will be viewable by the user, sorted by the nearest date
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Frequently, predecessor for ride sharing
Business Rules:	None
Special Requirements:	There are events to be viewed, otherwise an empty list will be displayed
Assumptions:	None
Notes and Issues:	None

3.14 Use Case 14: Viewing Ministries

Use Case ID:	14
Use Case Name:	Viewing Ministries
Created By:	Eric Tran
Last Updated By:	Eric Tran
Date Created:	October 25, 2015
Date Last Updated:	October 25, 2015
Actors:	One student interested in Cru/already involved with Cru
Description:	A user opens the app and wishes to view all possible ministries.
Preconditions:	<ol style="list-style-type: none"> 1. The app is already installed on student's iPhone 2. The user has WiFi / 4G connectivity
Postconditions:	<ol style="list-style-type: none"> 1. Student has knowledge of all ministries

Normal Flow:	1.0 Viewing Ministries <ol style="list-style-type: none"> 1. Student opens the app 2. Student navigates to “Get Involved” 3. Students views all ministries 4. Student is interested in a ministry. 5. Student opens up more information about specific ministry.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Every so often, when user would like to know more about other ministries
Business Rules:	None
Special Requirements:	Ministries informations will be present in the application
Assumptions:	None
Notes and Issues:	None

3.15 Use Case 15: Viewing Ride Status As A Driver

Use Case ID:	15
Use Case Name:	Viewing Ride Status As A Driver
Created By:	Tammy Kong
Last Updated By:	Tammy Kong
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One student already involved with Cru
Description:	A user wants to check who he/she is giving a ride to. The user will be able to view that information through that status section of ride share.
Preconditions:	<ol style="list-style-type: none"> 1. The app is already installed on student’s iPhone 2. The user has WiFi / 4G connectivity 3. The user is signed up as a driver for an event

Postconditions:	<ol style="list-style-type: none"> 1. The driver will be able to see if his/her car is full. 2. The driver will be able to see who he/she is giving a ride to.
Normal Flow:	<p>1.0 Viewing Ride Status as a Driver</p> <ol style="list-style-type: none"> 1. Member opens the Cru app 2. Member selects the Ride Share tab, which is in the same tab as Events 3. Member selects "Status" 4. Member will see which events he/she is offered to give a ride for 5. Member will see who is in his/her car.
Alternative Flows:	None
Exceptions:	<p>1.0.E.1 Member is not a driver</p> <ol style="list-style-type: none"> 1. Member did not sign up as a driver
Includes:	None
Priority:	Medium
Frequency of Use:	Couple hundred times per event
Business Rules:	None
Special Requirements:	None
Assumptions:	User is signed up as a driver for an event.
Notes and Issues:	None

3.16 Use Case 16: Viewing Ride Status As A Passenger

Use Case ID:	16
Use Case Name:	Viewing Ride Status As A Passenger
Created By:	Tammy Kong
Last Updated By:	Tammy Kong
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One student already involved with Cru

Description:	A user wants to check who he/she will be getting a ride from. The user will be able to view that information through that status section of ride share.
Preconditions:	<ol style="list-style-type: none"> 1. The app is already installed on student's iPhone 2. The user has WiFi / 4G connectivity 3. The user is signed up as passenger for an event
Postconditions:	<ol style="list-style-type: none"> 1. The passenger will be able to see the date, time, location of the ride 2. The passenger will be able to see the name and number of driver.
Normal Flow:	<p>1.0 Viewing Ride Status as a Passenger</p> <ol style="list-style-type: none"> 1. Member opens the Cru app 2. Member selects the Ride Share tab, which is in the same tab as Events 3. Member selects "Status" 4. Member will see which events he/she is getting a ride for 5. Member will see the information of his/her driver.
Alternative Flows:	None
Exceptions:	<p>1.0.E.1 Member is not a passenger</p> <ol style="list-style-type: none"> 1. Member did not sign up as a passenger
Includes:	None
Priority:	Medium
Frequency of Use:	Couple hundred times per event
Business Rules:	None
Special Requirements:	None
Assumptions:	User is signed up as a passenger for an event.
Notes and Issues:	None

3.17 Use Case 17: Receiving Notification of an Event

Use Case ID:	17
Use Case Name:	Receiving Notification of an Event

Created By:	Jordan Tang
Last Updated By:	Jordan Tang
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One student involved with Cru
Description:	User receives a push notification for immediate upcoming events
Preconditions:	1. User has the Cru app downloaded to their phone
Postconditions:	1. User has received a push notification on their phone
Normal Flow:	1.0 Receiving Notification of an Event 1. User receives notification on their phone
Alternative Flows:	None
Exceptions:	None
Includes:	Events
Priority:	High
Frequency of Use:	Low usage
Business Rules:	None
Special Requirements:	None
Assumptions:	There are events to be pushed
Notes and Issues:	None

3.18 Use Case 18: Viewing Guides

Use Case ID:	18
Use Case Name:	Viewing Guides
Created By:	Tammy Kong
Last Updated By:	Tammy Kong
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One student involved with Cru
Description:	The user wants to learn more information about their faith

Preconditions:	<ol style="list-style-type: none"> 1. The app is installed on the user's phone 2. The user has Wi-Fi / 4G connection
Postconditions:	<ol style="list-style-type: none"> 1. The user is able to find out more information about their faith
Normal Flow:	<p>1.0 Viewing Guides</p> <ol style="list-style-type: none"> 1. Member opens up the Cru app 2. Member selects the Resource tab 3. Member selects the guide tab 4. Member selects on a link they are interested in 5. The information will be displayed on the app
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Every so often, when user would like more information about their faith
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.19 Use Case 19: Viewing Articles and Studies

Use Case ID:	19
Use Case Name:	Viewing Articles and Studies
Created By:	Tammy Kong
Last Updated By:	Tammy Kong
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One student involved with Cru
Description:	The user wants to read articles and studies about their faith

Preconditions:	<ol style="list-style-type: none"> 1. The app is installed on the user's phone 2. The user has Wi-Fi / 4G connection
Postconditions:	<ol style="list-style-type: none"> 1. The user is able to find out more information about their faith
Normal Flow:	<p>1.0 Viewing Articles and Studies</p> <ol style="list-style-type: none"> 1. Member opens up the Cru app 2. Member selects the Resource tab 3. Member selects the "articles & studies" tab 4. Member selects on a link they are interested in 5. The information will be displayed on the app
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Every so often, when user would like more information about their faith
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.20 Use Case 20: Viewing Other Sites

Use Case ID:	20
Use Case Name:	Viewing Other Sites
Created By:	Jordan Tang
Last Updated By:	Jordan Tang
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One student interested or involved in Cru
Description:	A user wants to learn more about Cru or how they can get more involved

Preconditions:	<ol style="list-style-type: none"> 1. The app is installed on the user's phone 2. The user has Wi-Fi / 4G connection
Postconditions:	<ol style="list-style-type: none"> 1. Student has knowledge of all ministries
Normal Flow:	<p>1.0 Viewing Other Sites</p> <ol style="list-style-type: none"> 1. Student opens the app 2. Student navigates to "Resources" tab 3. Students selects the "Other Sites" tab 4. Student selects on a site they are interested in 5. The site will be displayed on the app
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Every so often, when user would like to know how to get more involved with Cru
Business Rules:	None
Special Requirements:	Other Sites will be present in the application
Assumptions:	There are sites to display
Notes and Issues:	None

3.21 Use Case 21: Syncing Academic Calendar

Use Case ID:	21
Use Case Name:	Syncing Academic Calendar
Created By:	Jordan Tang
Last Updated By:	Jordan Tang
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One student interested or involved in Cru
Description:	A user wants to sync their school's academic calendar
Preconditions:	<ol style="list-style-type: none"> 1. The app is installed on the user's phone 2. The user has Wi-Fi / 4G connection
Postconditions:	<ol style="list-style-type: none"> 1. The student's academic calendar is synced

Normal Flow:	1.0 Syncing Academic Calendar 1. Student opens the app 2. Student navigates to “My School” tab 3. Students selects the “Academic Calendar” tab 4. Student “Add to Phone Calendar” 5. Confirmation will be displayed
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	Low
Frequency of Use:	Probably once a year, every time there’s a new school calendar
Business Rules:	None
Special Requirements:	There are events to sync
Assumptions:	School has a calendar to sync from
Notes and Issues:	None

3.22 Use Case 22: Unsyncing Academic Calendar

Use Case ID:	22
Use Case Name:	Unsyncing Academic Calendar
Created By:	Jordan Tang
Last Updated By:	Jordan Tang
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One student interested or involved in Cru
Description:	A user wants to unsync their school’s academic calendar
Preconditions:	1. The app is installed on the user’s phone 2. The user has Wi-Fi / 4G connection
Postconditions:	1. The student’s academic calendar is unsynced
Normal Flow:	1.0 Unsyncing Academic Calendar

	<ol style="list-style-type: none"> 1. Student opens the app 2. Student navigates to “My School” tab 3. Students selects the “Academic Calendar” tab 4. Student “Remove from Phone Calendar” 5. Confirmation will be displayed
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	Low
Frequency of Use:	Probably once a year, every time there’s a new school calendar
Business Rules:	None
Special Requirements:	The app is must be able to cross reference events on calendar and events from the school calendar
Assumptions:	School calendar has already be synced
Notes and Issues:	None

3.23 Use Case 23: Viewing Campus Map

Use Case ID:	23
Use Case Name:	Viewing Campus Map
Created By:	Jordan Tang
Last Updated By:	Jordan Tang
Date Created:	October 27, 2015
Date Last Updated:	October 27, 2015
Actors:	One student interested or involved in Cru
Description:	A user who wants to view a map of their campus
Preconditions:	<ol style="list-style-type: none"> 1. The app is installed on the user’s phone 2. The user has Wi-Fi / 4G connection
Postconditions:	<ol style="list-style-type: none"> 1. The student is able to view the campus map
Normal Flow:	1.0 Viewing Campus Map <ol style="list-style-type: none"> 1. Student opens the app 2. Student navigates to “My School” tab 3. Students selects the “Campus Map” tab 4. App will display campus map

Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	Low
Frequency of Use:	Probably a couple of times for new members trying to find event locations
Business Rules:	None
Special Requirements:	None
Assumptions:	The school has a campus map available to be scrapped
Notes and Issues:	None

3.24 Use Case 24: Choosing Initial Campus and Ministry

Use Case ID:	24
Use Case Name:	Choosing Initial Campus and Ministry
Created By:	Tammy Kong
Last Updated By:	Tammy Kong
Date Created:	November 21, 2015
Date Last Updated:	November 21, 2015
Actors:	One student interested or involved in Cru
Description:	A user who wants to use the iOS application
Preconditions:	<ol style="list-style-type: none"> 1. The app is already installed on students iPhone 2. The student has not used the iOS application beforehand
Postconditions:	<ol style="list-style-type: none"> 1. The student will be able to know about events and resources from the selected ministry
Normal Flow:	1.0 Choosing Initial Campus and Ministry

	<ol style="list-style-type: none"> 1. Student opens the app 2. App will prompt student to select at least one campus 3. Students selects at least one campus 4. App will prompt student to select at least one ministry 5. Student selects at least one ministry 6. App will display the main screen of the app
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	High
Frequency of Use:	Every time a student downloads and opens the application for the first time
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

4 System Features

4.1 Ride Sharing

4.1.1 Description and Priority

Ride Sharing is a feature that will be an option for an event. If this feature is turned on, members of Cru will be able to offer a ride to other Cru members and people will be able to request a ride as well. The default pick-up/drop-off area will be in front of the PAC at Cal Poly, but event administrators can make this location more flexible. We will create an algorithm to efficiently pair up drivers and passengers based on their schedules and possible pick-up/drop-off location. Current Cru staff find it difficult and time consuming to plan ridesharing manually and the purpose of this feature is to help alleviate some of those difficulties. Ride sharing is of medium priority. Although its benefits to Cru are high, this feature is not detrimental to the overall functionality of the app.

4.1.2 Stimulus/Response Sequences

Stimulus:	Offer a ride
Response:	System displays questionnaire to figure out the driver's schedule

Stimulus:	Request a ride
Response:	System displays questionnaire to figure out user's schedule and assign to an available driver

Stimulus:	Check status of a ride
Response:	System will display whether their ride is pending or has been assigned

Stimulus:	Driver cancels ride offer
Response:	System will notify passengers and their ride status will be pending

Stimulus:	Passenger cancels ride request
Response:	System will notify driver that a passenger cancelled and one more seat is available

4.1.3 Functional Requirements

FR-1	Any app user subscribed to an event can request rides to that event
FR-2	Only registered Cru members can offer rides to events
FR-3	Driver will get notified when car is filled
FR-4	Driver and passengers will receive each others contact information
FR-5	Passenger and Driver can cancel their ride offer/request
FR-6	Option for one way or both way rides

4.2 Event Subscriptions

4.2.1 Description and Priority

Being able to receive event notifications will be of high priority for this application. Students will decide which ministry or ministries they would like to

join and be automatically subscribed to those events. A subscription entails the option of automatic push notifications to remind the user of an event. This is of high priority because the main purpose of the app is to get more involvement within Cru and reminding people of the upcoming events is a very important way of doing so.

4.2.2 Stimulus/Response Sequences

Stimulus:	Selects/changes ministry and is subscribed to their events
Response:	System displays the list of upcoming events based on their ministry selection

4.2.3 Functional Requirements

FR-7	User will see upcoming events based on campus and ministry
FR-8	Users can join and subscribe to multiple ministries
FR-9	User can subscribe to Crus Google Calendar and integrate it with the iOS native calendar app
FR-10	User can change the ministries they are subscribed for
FR-11	Push Notifications dependent on predetermined and admin created reminders

4.3 Cru Resources

4.3.1 Description and Priority

The Cru Resources tab provided a various amount of information such as the most recent YouTube videos, articles involving the Cru community, and links to social media accounts. Through this tab, admin users will be able to login to view any hidden resources. This feature is of medium priority. This will be a nice feature to help people stay updated with what is going on with Cru, but does not directly reach out to members of Cru to encourage people to get more involved with the community.

4.3.2 Stimulus/Response Sequences

Stimulus:	View YouTube videos
Response:	System displays list of most recent Cru YouTube videos

Stimulus:	View guides on faith, articles, and other links
Response:	System displays list of these available resources

Stimulus:	Login in as Cru Leader
Response:	System display hidden resources based on their login

4.3.3 Functional Requirements

FR-12	Automatically update the video archive of Cru meetings
FR-13	User will be able to filter out videos they are looking for by using the Cru video tags
FR-14	User will view links and articles by topic
FR-15	Login system for Community Group leaders to view hidden resources

4.4 Get Involved

4.4.1 Description and Priority

This app will help people get more involved in 2 ways: providing information about becoming part of a ministry team and determining a community group for the user to join. If the user wishes to join a community group, the system displays a questionnaire to determine which community group best fits their schedule and other needs. If the user wishes to become part of a ministry team, more information about this can be viewed through this section as well. This part of the app will also allow users to change their ministry. This feature will be of high priority because it focuses on expanding the Cru community.

4.4.2 Stimulus/Response Sequences

Stimulus:	Select a community group
Response:	System displays a questionnaire based on the ministry or ministries that the user has chosen to determine which community group best fits with them

Stimulus:	Select a ministry team
Response:	System sends user's info to the team leader

Stimulus:	Change ministry
Response:	System displays list of ministries to choose from based on the user's campus

4.4.3 Functional Requirements

FR-16	User will be able to join Community Groups through questions (name, gender, year, and other specific ministry questions)
FR-17	Filtration system to narrow down list of possible community groups based on questionnaire from REQ-1 (prioritized by smallest community group)
FR-18	Users will view list of Ministry teams they can get involved in
FR-19	Users can join multiple community groups based on all their ministries
FR-20	User can change Community Groups
FR-21	Community Group Leaders will be notified when someone selects their group

4.5 Summer Missions

4.5.1 Description and Priority

This section of the app will provide information about summer trips that are involved with the Cru community. This is of high priority since it provides information for members to get further involved with Cru.

4.5.2 Stimulus/Response Sequences

Stimulus:	Select a summer mission
Response:	System sends user to an outside link to apply for a summer mission

4.5.3 Functional Requirements

FR-22	User can see list of available summer missions
FR-23	Capability to click on trip for more information (dates, location, description, leaders, cost, picture)
FR-24	Link out to application for that trip

4.6 My School

4.6.1 Description and Priority

This tab will have information about the Campus the user is involved with. This includes the school's academic calendar, a campus map, and the option for the user to change their campus. This feature is of low priority since this does not directly help students get more involved with Cru. Also, which school information getting displayed may get complicated with users who are involved with multiple campuses.

4.6.2 Stimulus/Response Sequences

Stimulus:	View the school's academic calendar
Response:	System displays the school's academic calendar

Stimulus:	View the school's campus
Response:	System displays the school's campus map

Stimulus:	Change campus
Response:	System displays list of available campuses to choose from

4.6.3 Functional Requirements

FR-25	Pull schools academic calendar
FR-26	Zoomable school building maps with names and building numbers
FR-27	Option to change users campus

5 External Interface Requirements

5.1 User Interfaces

The user interface of the Cru Central Coast app will adhere to the Cru Branding style guidelines for Cru Central Coast. Doing so will ensure that we keep their brand strong and reinforced. The UI will fit to the screen size of each individual hardware supported by the app, listed under 5.2.

The Cru Central Coast app will be divided into 5 tabs: “Events”, “Resources”, “Get Involved”, “Summer Missions”, and “My School”, with “RideSharing” being integrated with “Events”.. Tap-based gestures will allow users to switch between the different tabs of the application, as well as accessing features within those tabs.

5.2 Hardware Interfaces

Supported device types for the Cru Central Coast app are iPhone 5, iPhone 5s, iPhone 6, iPhone 6s, iPhone 6+, and iPhone 6s+.

5.3 Software Interfaces

The supported operating system the Cru Central Coast app uses is iOS. The minimum iOS version we will support is iOS 9. The app will be written in Swift using Xcode 7.0.1. External libraries that will be used are Google Calendar API and YouTube iOS API.

5.4 Communications Interfaces

The communication functions required by the Cru Central Coast application will be electronic forms for identifying and determining the Community Group options for a particular user. Electronic forms will also be used to determine best possible drivers for the user in RideSharing. In addition, the process of joining a ministry team and applying for a summer mission will require the user to go through an external sign up page.

The Cru Central Coast application will communicate with the Cru MongoDB Database through a REST api we will have made. Communication with the database will include read-enabled and write-enabled activities.

6 Other Nonfunctional Requirements

6.1 Performance Requirements

PR-1	The system shall not take longer than 10 seconds to obtain and load information from the database unless database is down.
PR-2	The system shall be able to handle a few hundred users concurrently.
PR-3	The system shall be able to automatically update new information once a user change their preferences.
PR-4	The system shall be able to detect quickly if a car is full when multiple members are signing up for rides.

6.2 Safety Requirements

SAFETY-1	This system shall not be intended in any way to harm any user.
SAFETY-2	The system shall verify if a potential driver is a Cru member before allowing user to offer rides to other members.

6.3 Security Requirements

SR-1	All preferences (selected colleges, selected ministries, questionnaires, etc.) set on the system shall only be stored into the user's device.
SR-2	The system shall require login for community leaders to view additional resources.
SR-3	The system shall have proper security for the login process.
SR-4	The system shall give out community leaders' information once the users have input their own information down.

6.4 Software Quality Attributes

SQA-1	All code written will follow Cru's Coding Style.
SQA-2	UI/UX design shall be made simple and easy for any users to navigate through the system.
SQA-3	The system shall be free of charge for all users.
SQA-4	The system shall be modularized for future extensions.
SQA-5	The system shall be able to supported on the iOS 9.