



CS577a 17Fall
Team 04

:: Outline

1. Operational Concept Overview
2. UI Demo
3. Test Cases and Results
4. Quality Focal Point
5. Transition Plan

:: Operational Concept Overview

::: Shared Vision: Program Diagram

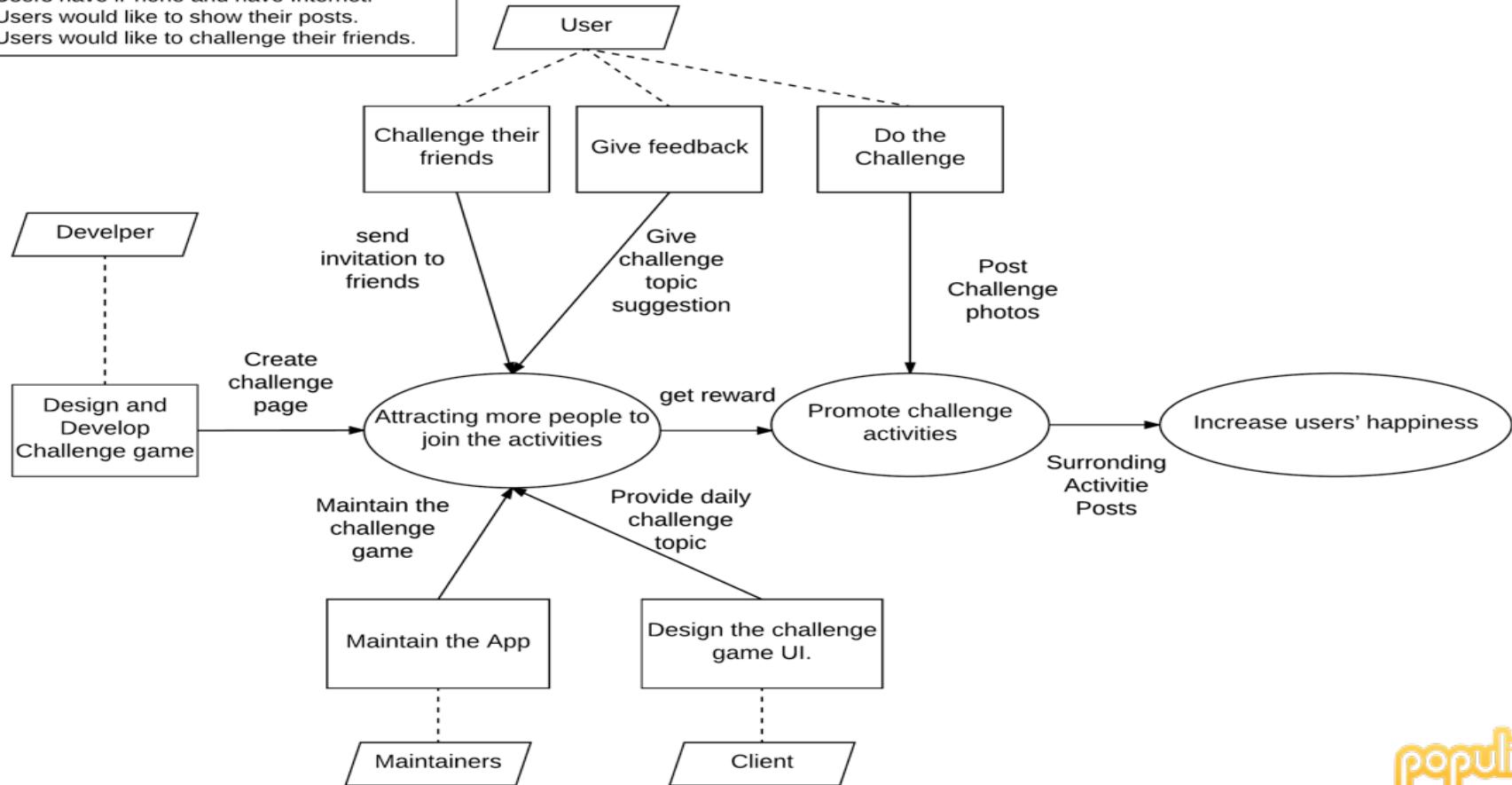
Assumptions:

- Users have iPhone and Internet
- User would like to challenge their friends
- User would like to show their posts

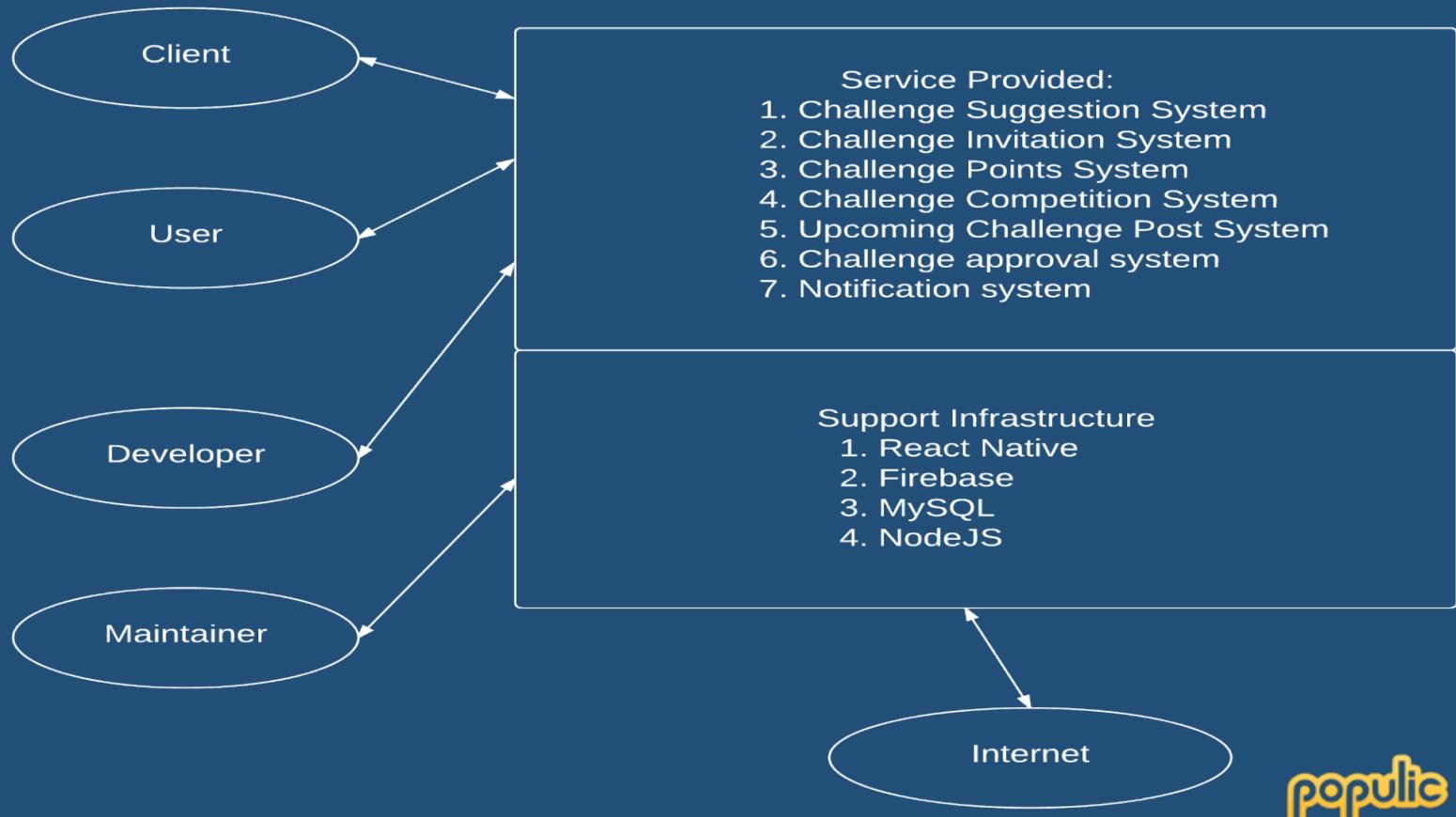
Stakeholders	Initiatives	Value Propositions	Beneficiaries
<ul style="list-style-type: none">• Developers• Client• Maintainers• Users	<ul style="list-style-type: none">• Develop the challenge part of the app• Design the app(UI), advertise the app• Maintain the app• Keep using the app and give feedback• Challenge their friends	<ul style="list-style-type: none">• Attracting more people to join the activities• To promote activities• To help students to find out what is happening around them and increase user's happiness	<ul style="list-style-type: none">• Users• Client

::= Benefit Chain Diagram

- Users have iPhone and have Internet.
- Users would like to show their posts.
- Users would like to challenge their friends.



::System Boundary & Environment



:: Core Capabilities

Capability Goals	Priority Level
OC-1 In-App friend List, Phone Contact List Access and SMS: The user can access the iphone contact list and send challenge invitation through iphone msg system, also can send invitation through notification to friends in-App.	MUST HAVE
OC-2 Challenge Complete Competition: The user can choose one friend to compete the time that they finished daily challenge	MUST HAVE
OC-3 Challenge Topic Suggestion: The user can send their feedback and challenge ideas to client	MUST HAVE
OC-4 Upcoming Challenge Post: The system will post upcoming 5 days daily challenge topic.	MUST HAVE
OC-5 View, Approve or Decline Challenge Post: The users are capable of viewing, approve or decline their friends challenge post.	MUST HAVE
OC-6 Challenge Game Pop Screen: The user will touch the pop screen to get all daily challenge information	MUST HAVE
OC-7 Offline & Online Notification: The user will send and get notification from populic.	MUST HAVE
OC-8 Competition Points: The user will get reward points from challenge competition	MUST HAVE

∴ Goals & Constraints

OG-1: Enlarge user's happiness.

OG-2: Raise interactions between different users.

OG-3: Increase the amount of active users in Populic.

CO-1: iOS as an Operating Systems.

CO-2: React Native as a development language for front end.

CO-3: Firebase as backend for application notification system.

CO-4: MySQL as backend for system.

... Demo

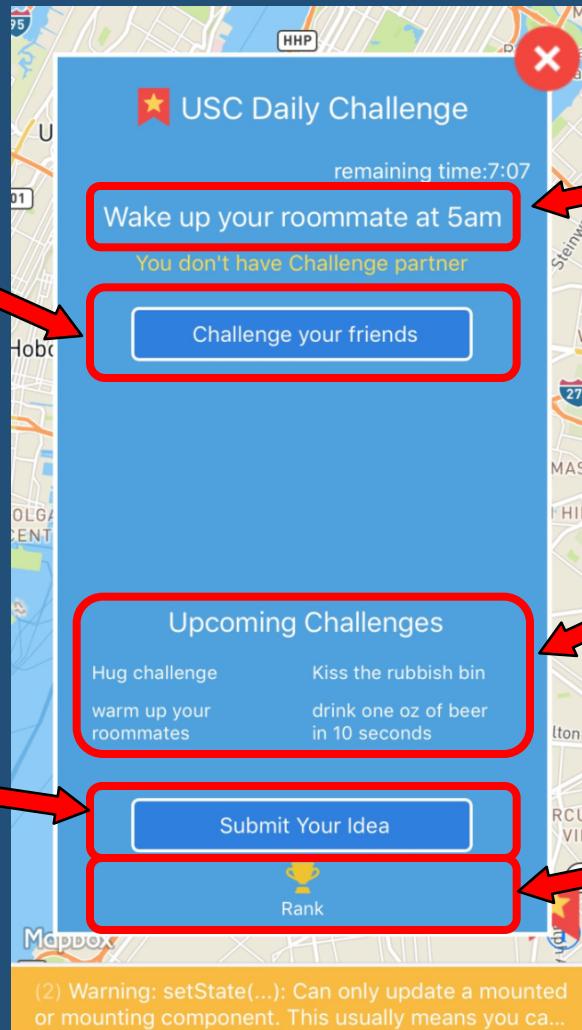
∴ Demo

1. Challenge Homepage Introduction
2. Challenge Others Page Introduction
3. Selecting Page Introduction
4. Competition Page Introduction

::: Homepage

Challenge Others

Today's challenge

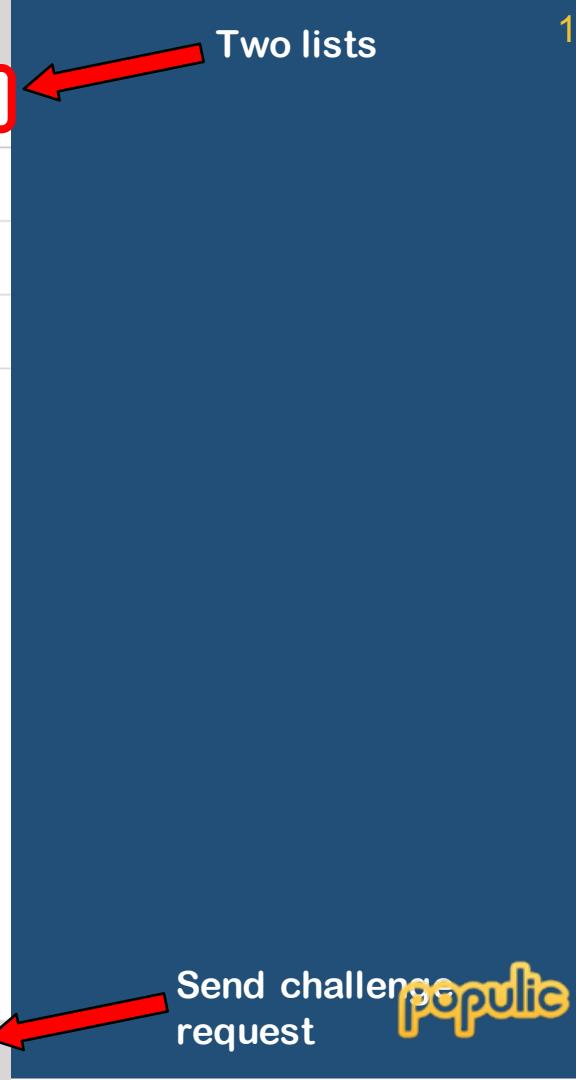
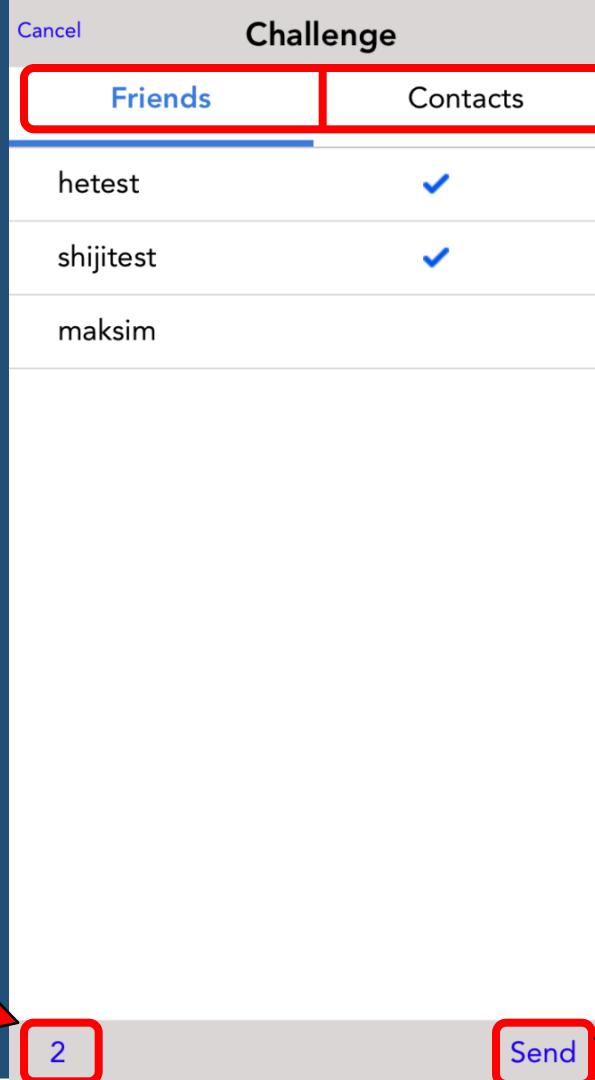


Provide idea

Upcoming challenge

Rank System

::: Challenge Others, cont



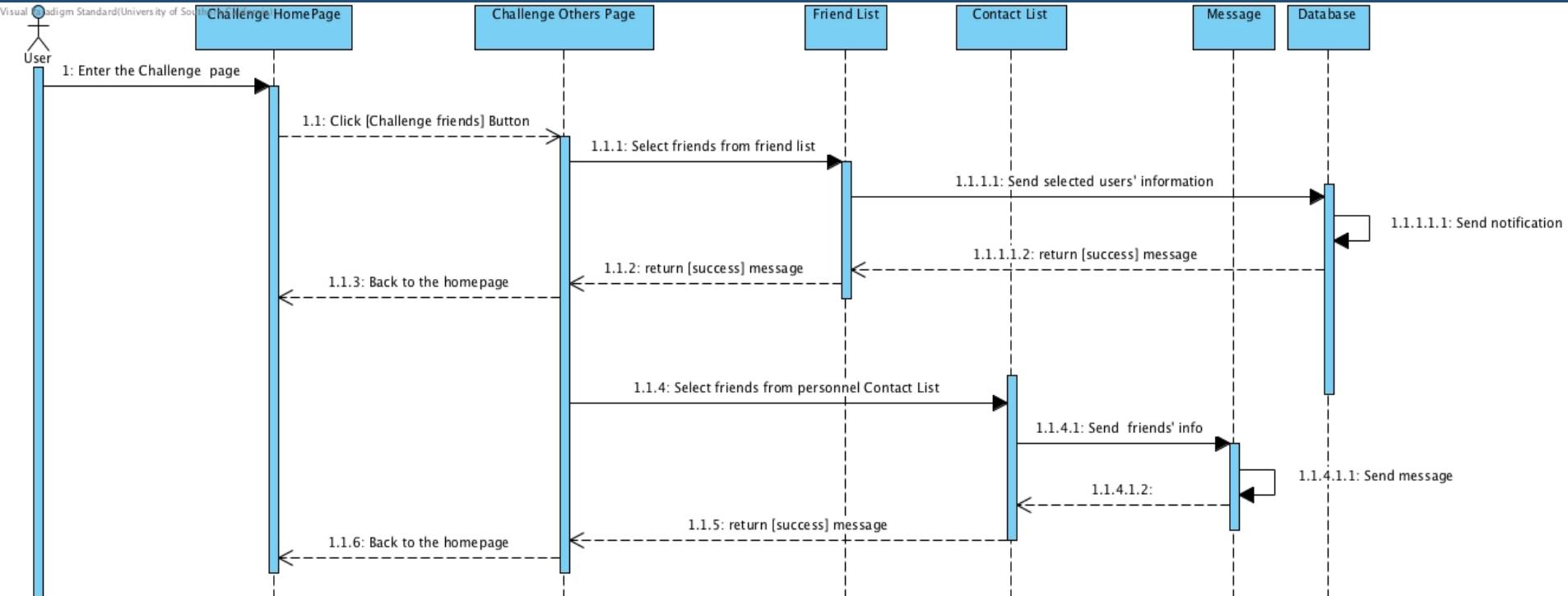
Total number of friends
you have challenged

2

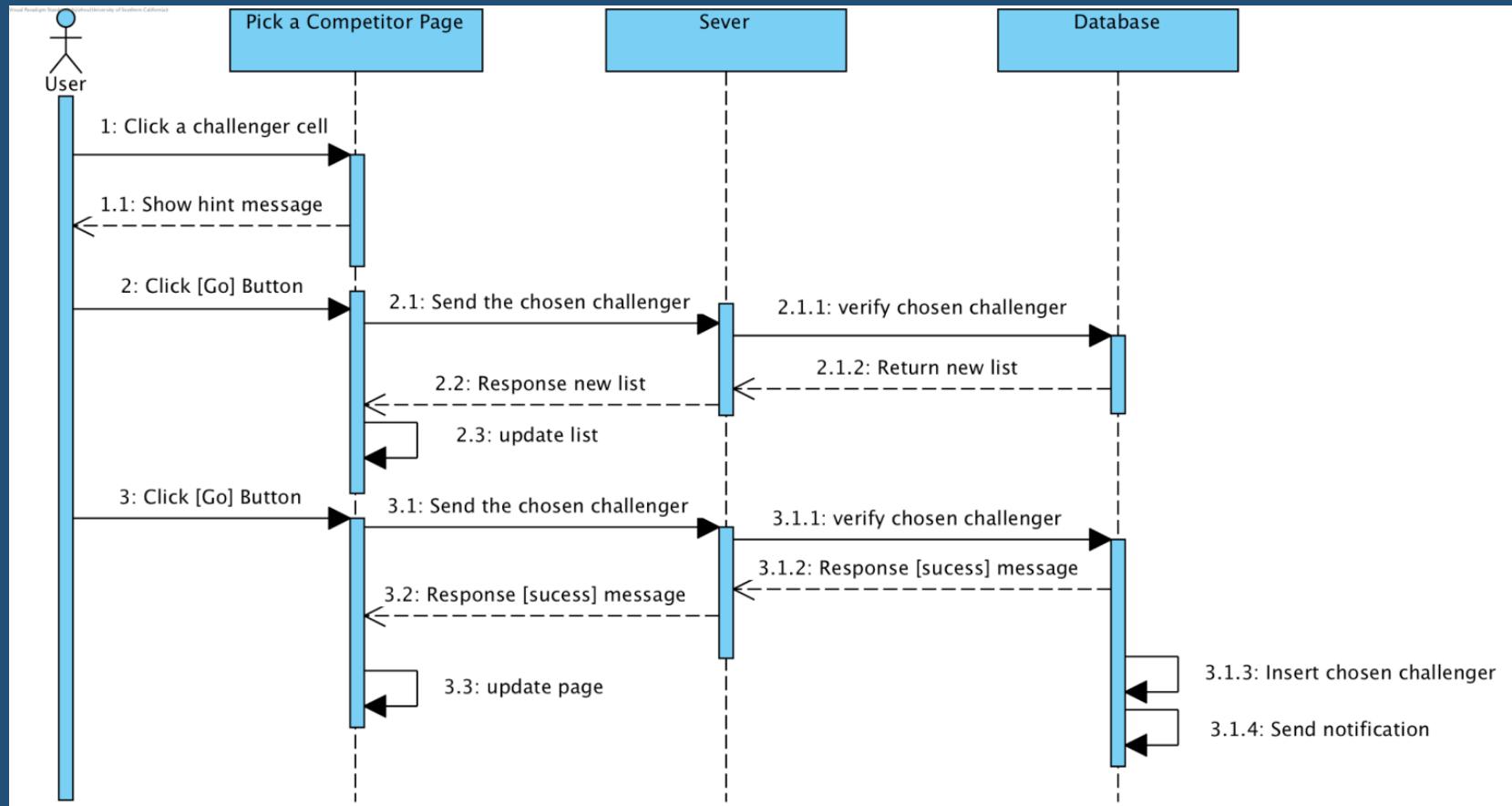
Send

Send challenge
request

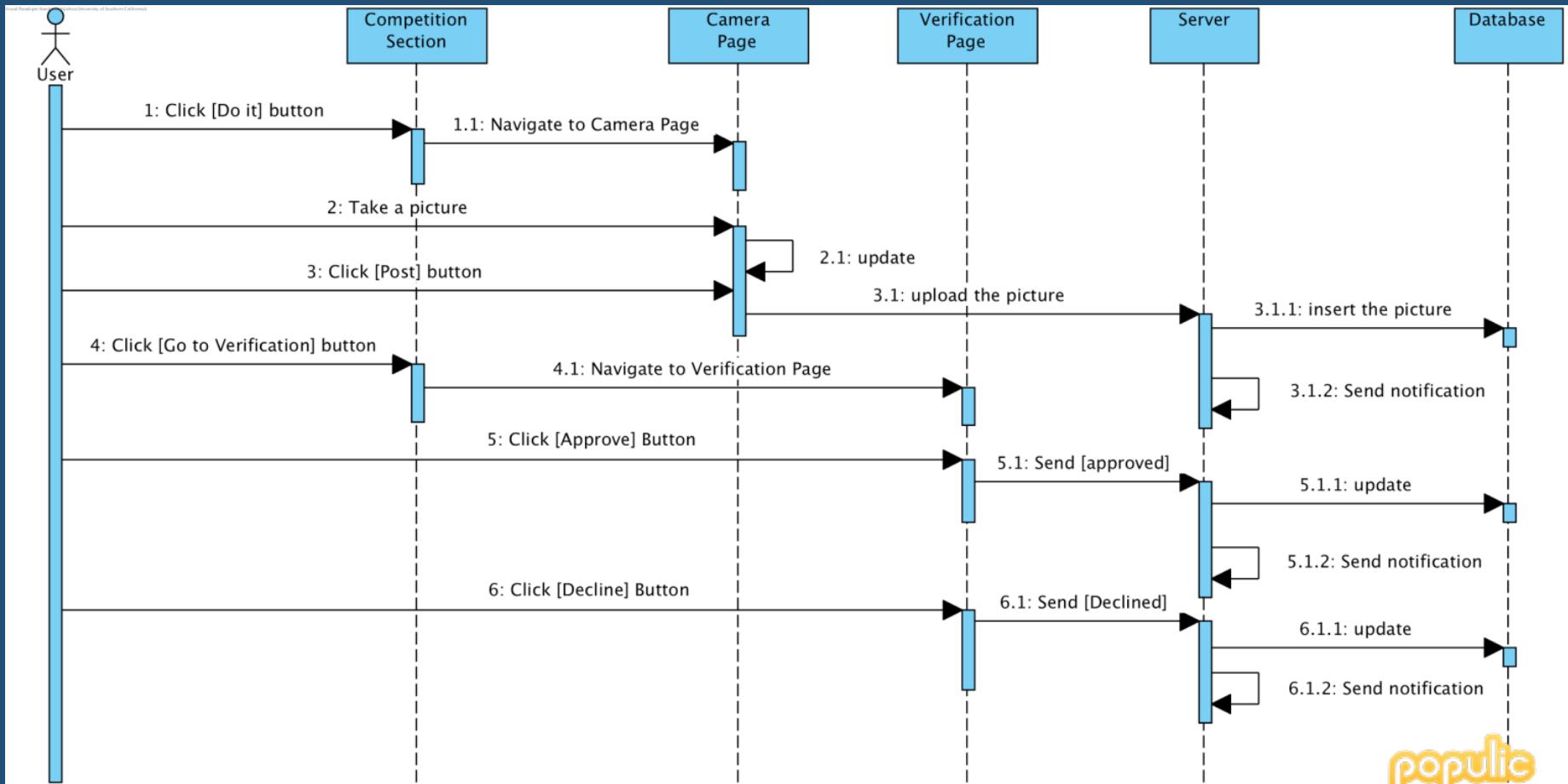
::: Challenge Others



::: Pick a Competitor



..: Competition Section



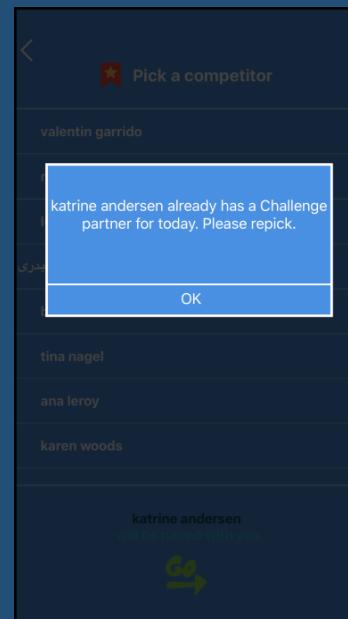
::: Pick a Competitor

Challenger List

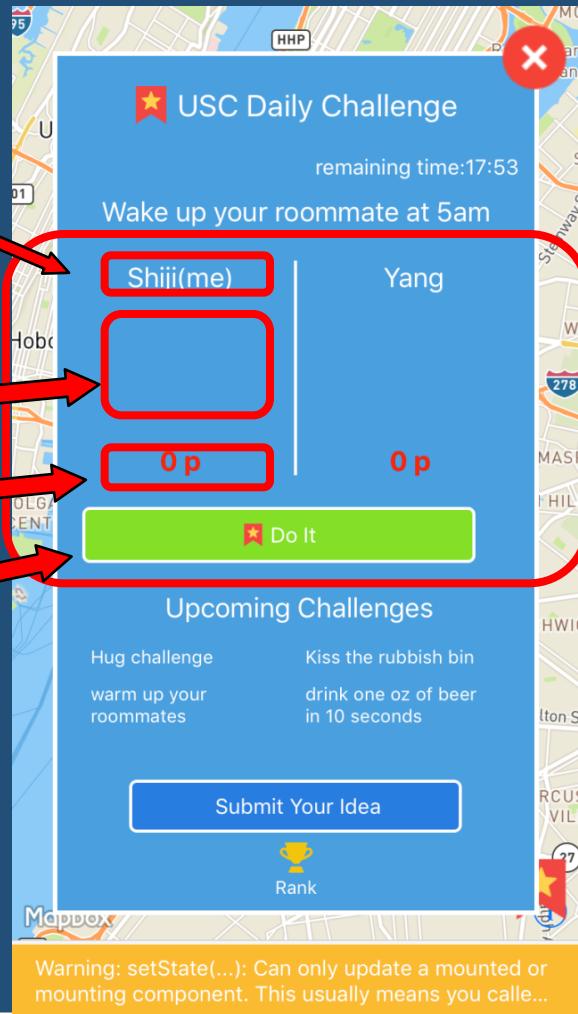


Selected item

Hint



:: Competition Section



Either you or your competitor has 4 status:

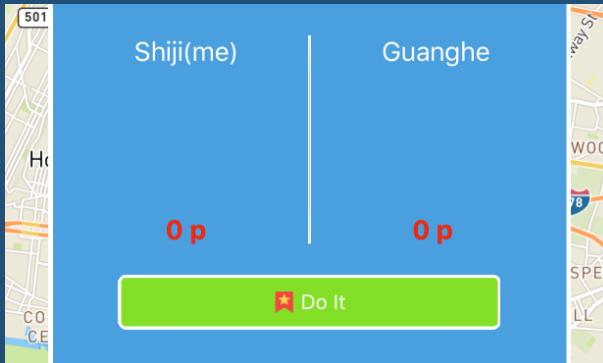
- Haven't done the challenge
- Complete & wait for verification
- Have been approved
- Have been declined

::: Competition Section

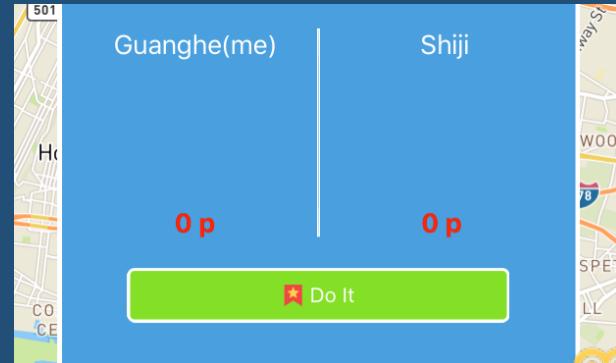
<i>status</i>	<i>isMatched</i>	<i>hasPost</i>	<i>isApproved</i>	<i>isDeclined</i>
1	False	-	-	-
2	True	False	-	-
3	True	True	False	False
4	True	True	True	False
5	True	True	False	True

- ← Haven't done the challenge
- ← Complete & wait for verification
- ← Have been approved
- ← Have been declined

In My App:



In My Competitor's App:



::: Test Cases and Results

- Purposes
 - 1. Check if the system meets the requirements
 - 2. Find faults as early as possible and eliminate them before put into daily use
- White box testing
 - 1. Use white box testing to validate and verify MySQL queries
- Black box testing
 - 1. Scrutinize whether the firebase can send notification correctly
 - 2. Use the black box testing to check the correctness of various functions, like: Challenging others, competitor selection, score system and rank system

:: TC-01 Navigation

21

Test Case Number	Test Item	Result
TC-01-01	The bottom right button in the home map page should link to the pop up challenge page.	Pass
TC-01-02	The today's challenge button should link to an extra pop up page in which the detail description of challenge is provided.	Pass
TC-01-03	The challenger button should navigate users to the competitor selection page.	Pass
TC-01-04	The submitting button should allow users to submit their own idea about challenge topic to the system.	Pass

:: TC-01 Navigation

22

Test Case Number	Test Item	Result
TC-01-05	The rank button in challenge homepage should demonstrate the rank of all users' scores in non-ascending order.	Pass
TC-01-06	The challenge friends button should have link to take users to the challenge others page, in which a user can challenge his/her friends from either friend list or contact list.	Pass
TC-01-07-01	Whenever the user has hitted the send button in challenge others page, if the challenged friends come from Populic, the system should navigate them back to the homepage automatically. If the challenged friends come from contact list, the system should navigate users to message page and allow them to send message.	Pass



:: TC-01 Navigation

Test Case Number	Test Item	Result
TC-01-08	Whenever the user picked up an available challenger as the competitor, the system should take this user to the competition page.	Pass
TC-01-09	Whenever the user picked up an unavailable challenger as the competitor, the system should illustrate a dialog to tell this user is unavailable, and then update the competitor candidate list automatically in the selection page.	Pass
TC-01-10	The do it button in competition page should link to the camera in which a user can take picture/shot a video.	Pass

:: TC-01 Navigation

24

Test Case Number	Test Item	Result
TC-01-11	Whenever the user send a challenge post successfully, the system will take this user back to competition page.	Pass
TC-01-12	The verification button allows a user to approve / disapprove his/her opponent post.	Pass
TC-01-13	Whenever a user finished the verification process, the system will take him / her back.	Pass
TC-01-14	The go back button in the top-left corner in every page should take a user back to previous page.	Pass

:: TC-02 Notification

25

Test Case Number	Test Item	Result
TC-02-01	The system should send challenge notification to a user if he / she has been challenged by others.	Pass
TC-02-02	The system should send competitor notification to a user if he / she has successfully been picked up as a competitor with one of his / her friends.	Pass
TC-02-03	The system should send approving notification to a user immediately after the opponent has completed the challenge.	Pass
TC-02-04	The system should send the approved / disapproved notification to a user as soon as his / her posts has been verified.	Pass

:: TC-02 Notification

26

Test Case Number	Test Item	Result
TC-02-05	The notification should be shown as a dialog at the top of the screen and is supposed to take the user back to our app when he / she clicks it.	Pass
TC-02-06	Whenever a user receives a notification, the icon should have a red spot at the top-right corner.	Pass
TC-02-07	The application should take users to Message page and fill in “To:”, textfield automatically when the user want to change friends in contact list.	Pass

:: TC-03 Competition logic

Test Case Number	Test Item	Result
TC-03-01	If both of two challengers are available (not matched), they are supposed to be matched successfully whenever one of them picks up another.	Pass
TC-03-02	If one of those two challengers is not available, the database should detect it and forbid the duplicated match.	Pass
TC-03-03	If two users have challenged the other simultaneously and one of them picked up another in advance, then the status for both of them should be matched.	Pass

:: Quality Focal Point

::: Traceability Matrix

Capability Goals	Requirements	Use Cases	Test Cases
OC-1 In-App friend List, Phone Contact List Access and SMS	WC_4596	UC-07	TC-01-07, TC-01-15, TC-01-16
OC-2 Challenge Complete Competition	WC_4596	UC-03, UC-05	TC-01-03, TC-03-02, TC-03-03
OC-3 Challenge Topic Suggestion	WC_4601	UC-06, UC-10	TC-01-04
OC-4 Upcoming Challenge Post	WC_4595	UC-02	TC-01-02

::: Traceability Matrix

Capability Goals	Requirements	Use Cases	Test Cases
OC-5 View Approve or Decline Challenge	WC_4537, WC_4600	UC-08	TC-01-06, TC-01-12
OC-6 Challenge Game Pop Screen	WC_4531	UC-01	TC-01-02
OC-7 Offline and Online Notifications	WC_4599	UC-03, UC-05	TC-02-01, TC-02-02, TC-02-03, TC-02-04, TC-02-05, TC-02-06
OC-8 Competition Points	WC_4598	UC-05	TC-01-05,

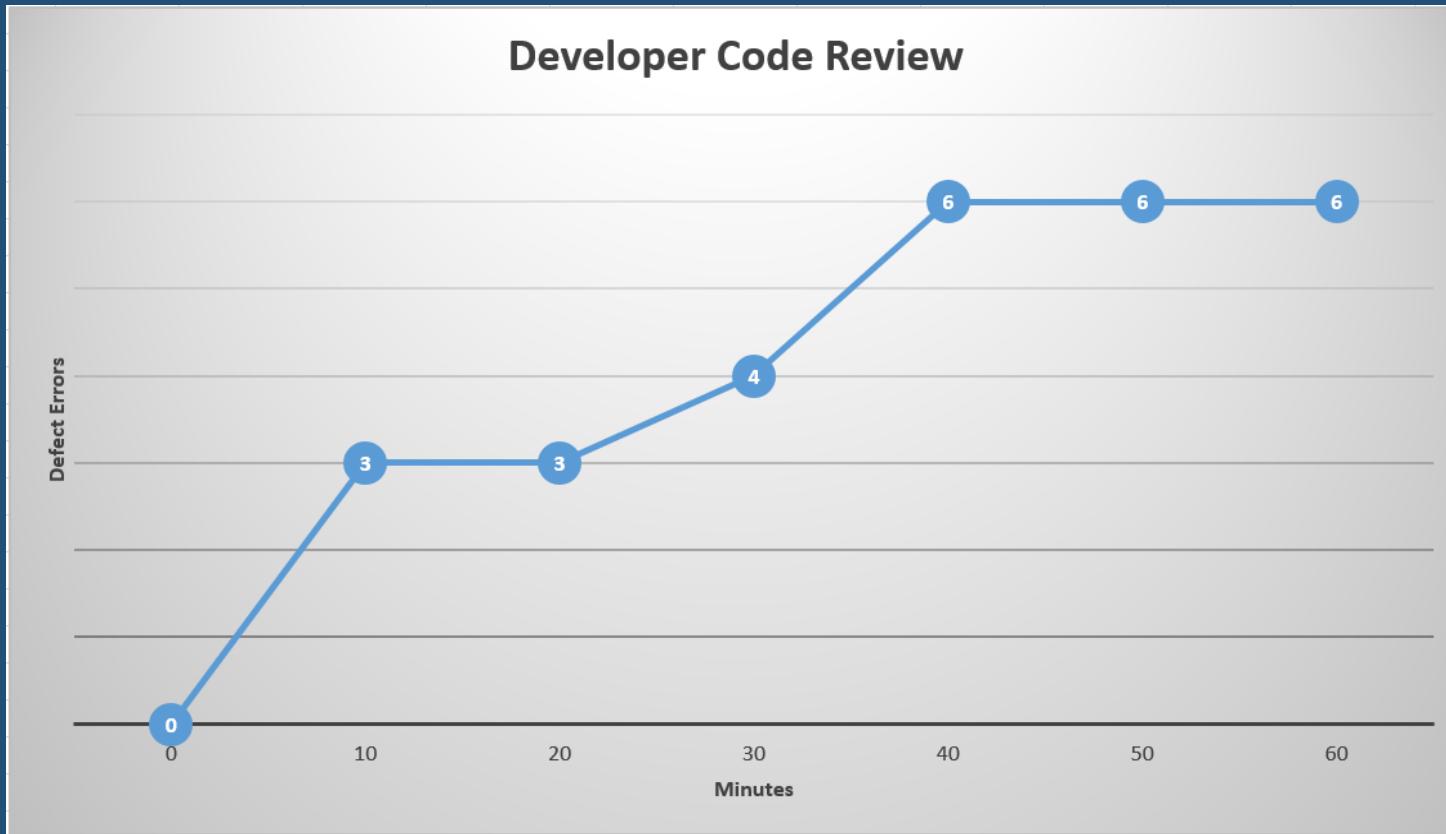
::: Technical Debt

- Solved
 - Slight inconsistencies between modules and coding styles. Solved: By implementing similar modular structure and file naming. For Push Notifications Modules, coding structures were refactored to mock existing code modules. Spent extra time into refactoring.
 - Parallel Development - Separation of development branches Solved: Coordination between developers and coordinating code reviews through meetings.
 - Lack Of Knowledge - Solved: Investing time to learn new technologies such as react native and redux.

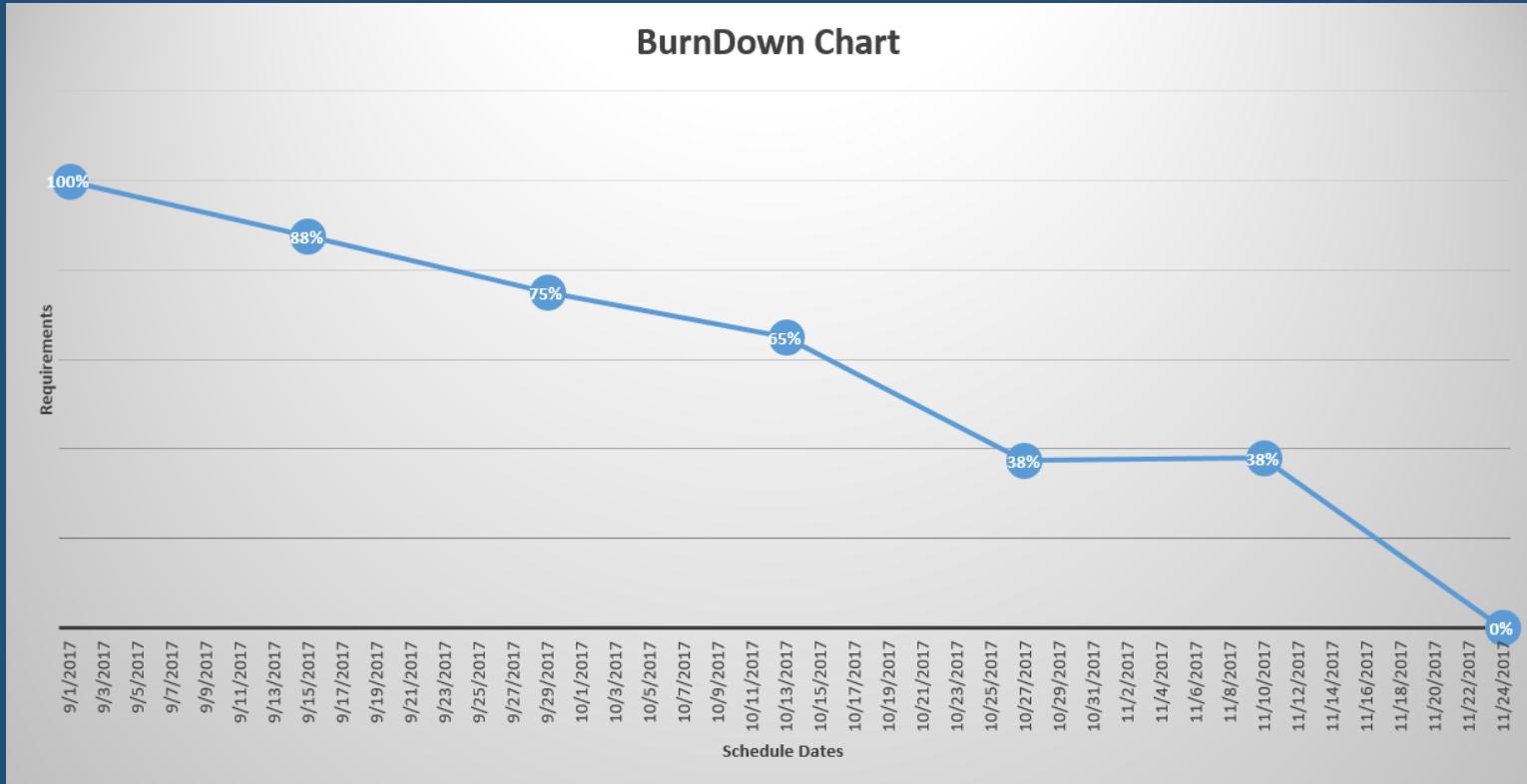
::: Technical Debt

- Remaining
 - Sending Text Messages From Backend Requires Alternative Method. Navigation to Messenger app. Text Message requires Sim Card to send.
 - Notifications(Game) - Implemented to meet requirements and be workable. Method of implementation disallows for further additional features associated with task (User Navigation to Corresponding Page)
 - Code Formatting - File Splitting (Backend Code) Takes time scrolling through file to find code segments
 - Lack of Alignment of Standards - Ignored Some slight standards for code in order to deliver workable project

::: Metric: Developer Code Review



::: Metric: BurnDown Chart



... Transition Plan

::: Objective of Transition Plan

- Get ready for transition and test
- Deploy our app on our client's server
- Test and evaluate app on our client's server
- Deliver all the source code and the document

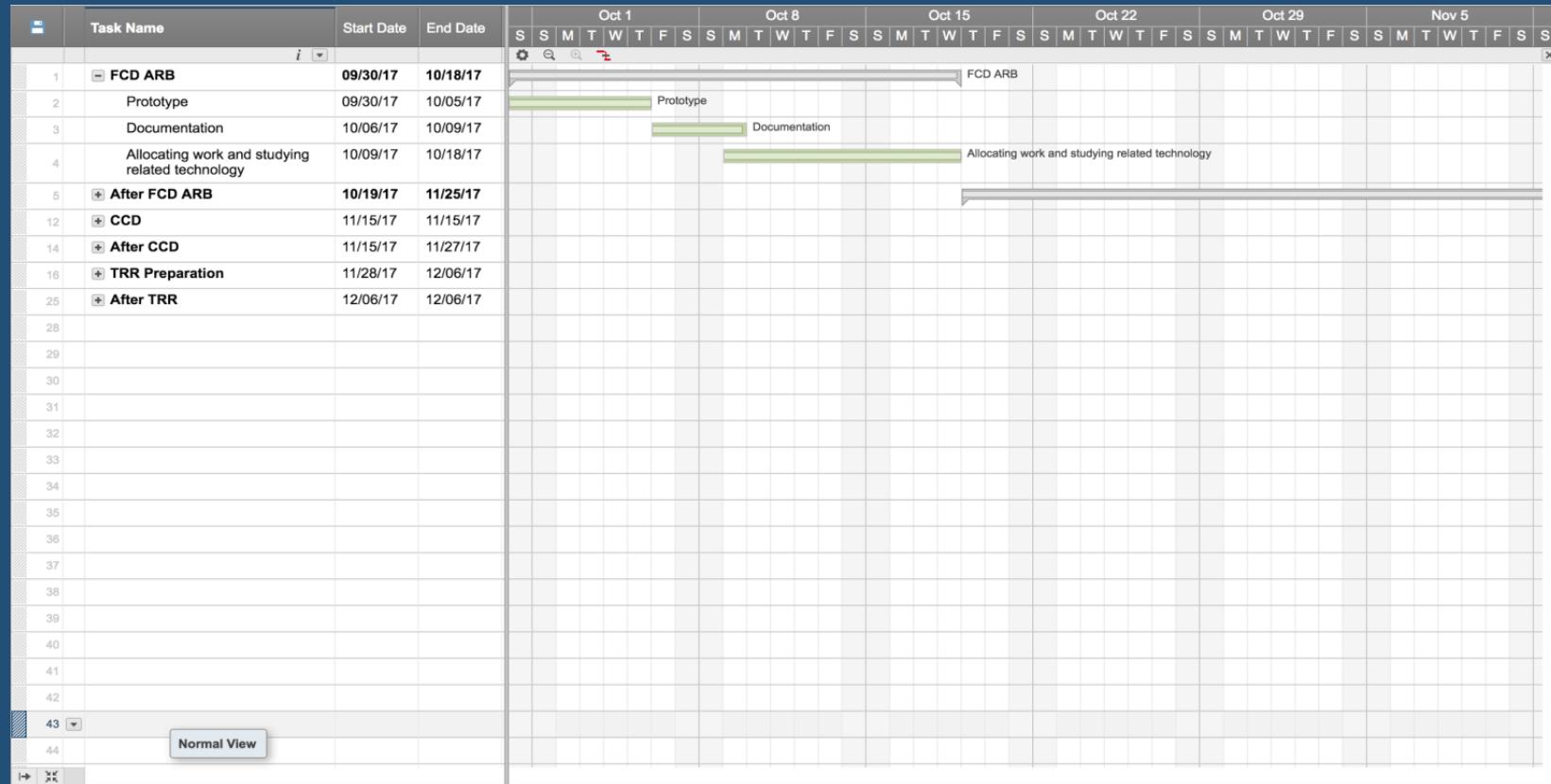
::: Preparation

Category	Item
Hardware	Macbook(several), iPhone(several)
Software	Xcode, Webstorm, React Native, Mysql, Firebase, Ubuntu14.0.4, Nodejs, GitHub
Site	On campus
Staff	Developer Team, Tester, Trainer, Client

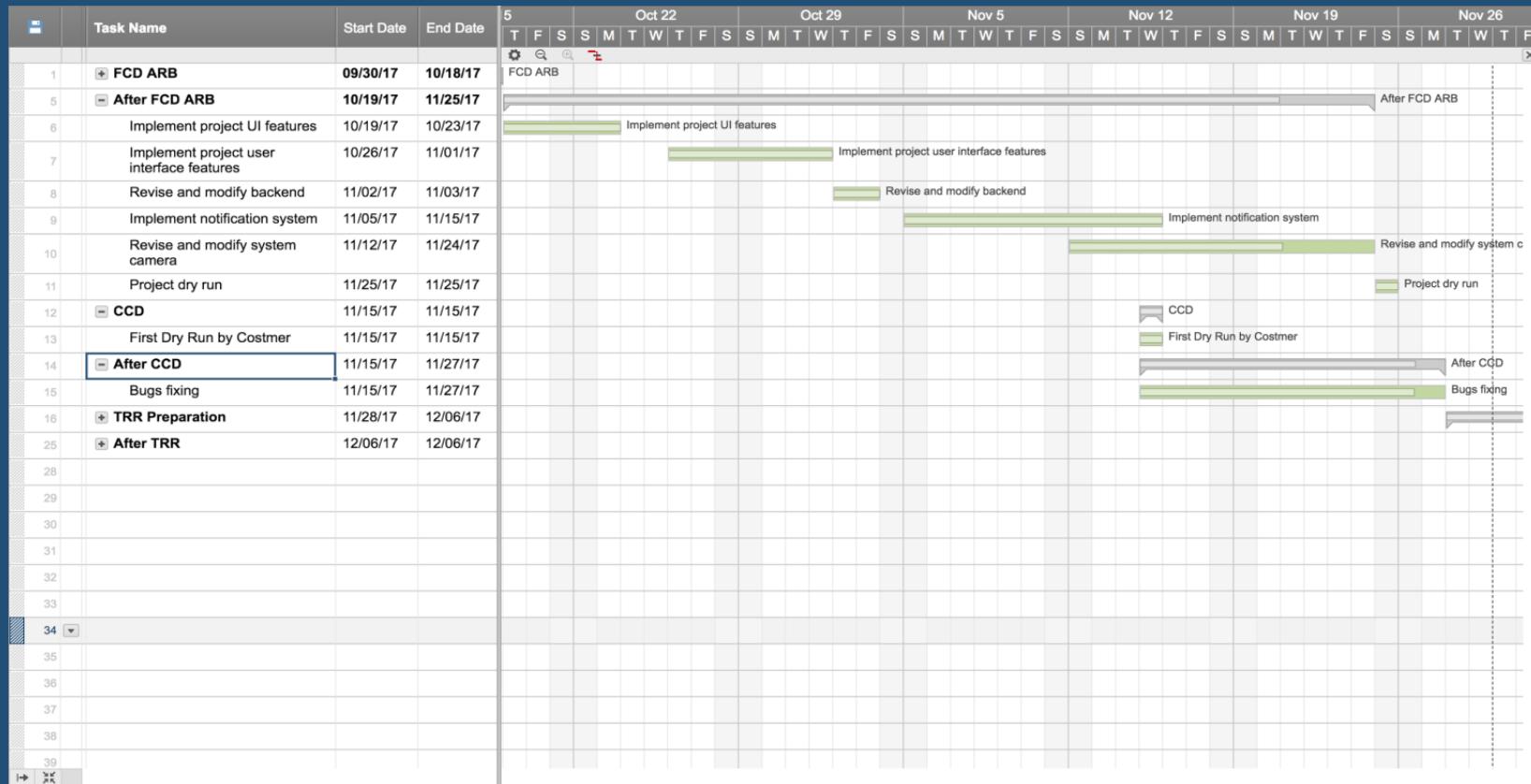
::: Operation

Category	Operation
Testing	Provide Testing Document and Acceptable Test Case (We have done beta test, find several bugs and fixed most)
Training	Provide Technical Support and Document (Our app is not obvious to use, but we write a easily readable manual for user to operate)
Evaluation	Client Feedback (We have received the beta test feedback and trying to satisfy all the reasonable requirement)

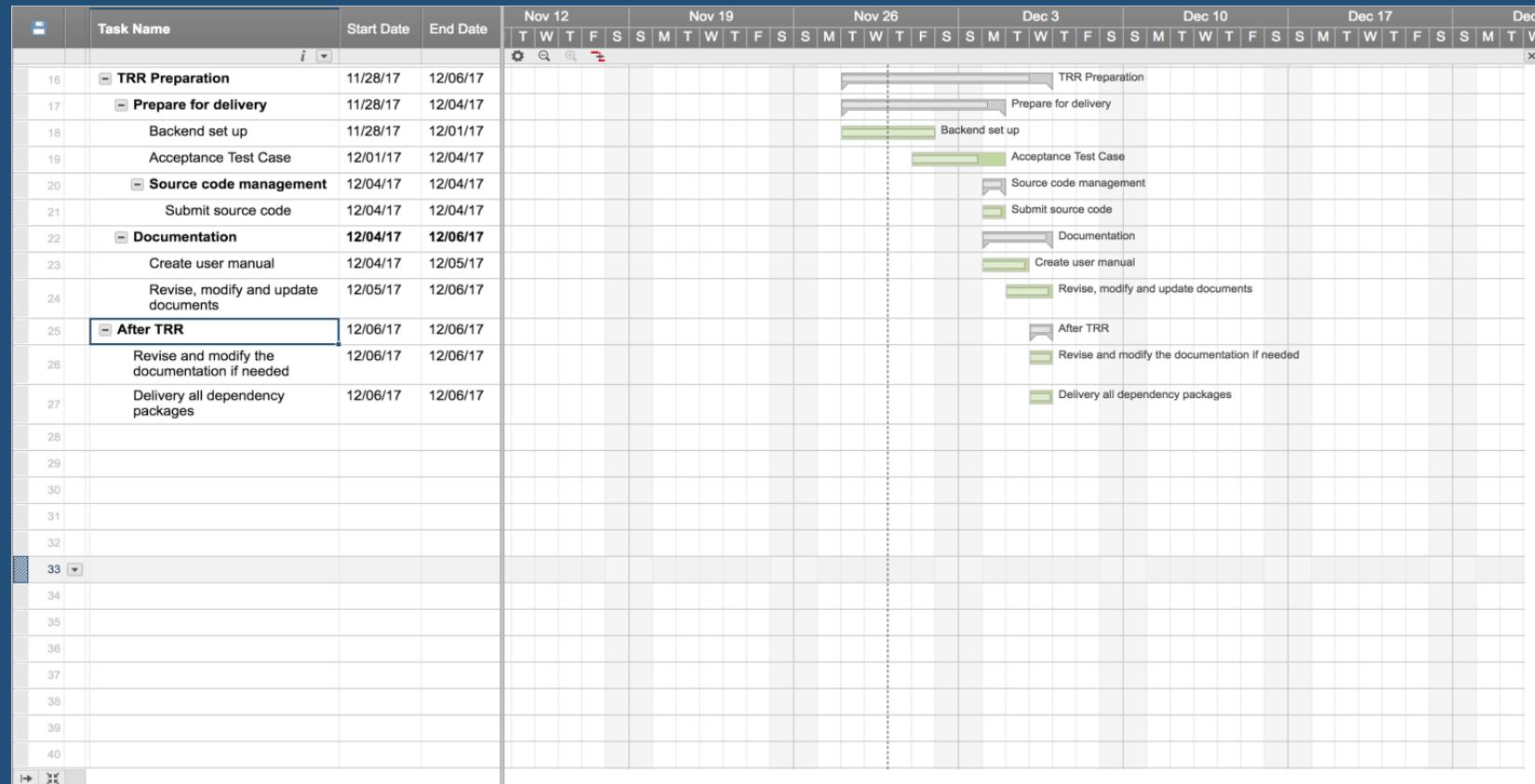
:: Milestones Plan



:: Milestones Plan



∴ Milestones Plan



:: Stakeholder Role & Responsibilities

Date	Role	Responsibility
12/01/17	client	Provide authorization of firebase and maintenance and update schedule
12/04/17	Front-end team	Fix bugs / Test / Provide technical support
12/04/17	Back-end team	Offer all the challenge API and database Schema.
12/06/17	Team	Have all documents ready and deliver the whole function
12/06/17	Maintainer	Learn to maintain upcoming challenge

::: Transition Risks

Maintenance

- Project meeting.
- Work breakdown document
- User manual
- Maintenance video

Ambiguous code

- refactor the code
- Finish the comment and document of our code

:: Required Resources

- Technical Manual
 - Guide on Setting up FireBase
 - DB Schema
 - Software Product Elements
 - Guide on how to manage Database.
- User Manual
 - Simple guide on how to use main features of challenge function.
 - Display of difference situation
- Source Code
 - Github

∴ Q & A