QuikPik
Sprint #1 Presentation

Anthony Pham, Jalon Flores, Juan Villa

Judy Tran, Ranjit John and Ernie Argel

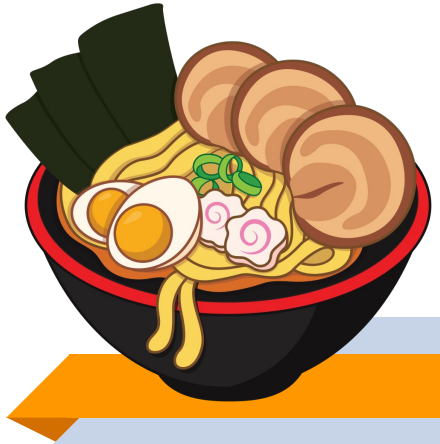
Product Requirement Document

Goals

Purpose: This application **saves the user time** and frustration of **deciding on a place to eat** by suggesting locations they would most likely enjoy **based on their defined preferences.**

Problems:

1. Too much time is spent deciding on a place to eat.
2. Customers are unsure what foods are available around them
3. Customers want to try new foods.



Goals (Cont.) - How does Quikpik streamline the current process?

Current process (using Yelp)

Users search for food keywords.

search through results to come to a decision based on user reviews, distance, price, and food type.

Streamlined Process

Users create a profile with preferred food types.

When they use Quikpik, the app will search for food locations based on their preferences and location in order to save time and make the process easier.

Personas (1 of 3)

Stefani Zhou, 28
Palo Alto, CA
Research Engineer

Background
Stephanie is a Senior Research Engineer and is responsible for managing a research team in hey company. She typically works with a 10-15 member team and is responsible for setting up end of the month lunch meetings with them.

Goals
Stefanie wants to have efficient but meaningful work be done at work. She wants her team to be engaged and always have high morale. She wants to have a great location for her meetings that can fulfill all of her goals.

Frustrations
Stefanie doesn't want to waste time by scrolling through Yelp's reviews to find a restaurant.

Motivations
Stefanie's main motivation is to have a change in scenery for her next team meeting.
She wants the location she picks to be both professional but also have some good food that all of her team members enjoy.

Personas (2 of 3)

David Stanley , 23
Santa Barbara, CA
College Student

Background
David is a full-time business student and works a part-time job at the local supermarket.

Goals
David wants to try new and unexplored restaurants near his college and home. He believes he has seen all the best restaurants around and is tired of all of them. David needs an app that shows him a new restaurant that caters to his food preference, which is affordable and nearby.

Frustrations
He hates to spend a lot of time planning with his indecisive friends on where they should eat.

Motivations
David's main motivation is to explore an eatery that he has never been before. He wants to become the 'cool' friend who knows of that great new hole in the wall restaurant that his friends never knew existed in their city.

Personas (3 of 3)

Stephen Clark, 35
Chicago, Illinois
Salesman

Background

Stephen is a salesman for Solar City . He loves to travel with his wife and try new cuisines. Stephen has visited over 15 countries and tried many types of exotic dishes.

Goals

Stephen wants to travel to places he has never been before and learn about the culture and people there. He loves to try new foods and often research about unique restaurants in the areas he visits.

Frustrations

Stephen hates going through the Yelp searches to find a restaurant. Often times, there won't be information about customer parking, rush hours, and etc.

Motivations

Stephen's main motivation is to enjoy his vacations without wasting time researching about a different restaurants he can visit.

Stephen wants a centralized source where all information about a restaurant(parking, hours of operation, menu and etc.) can be found.

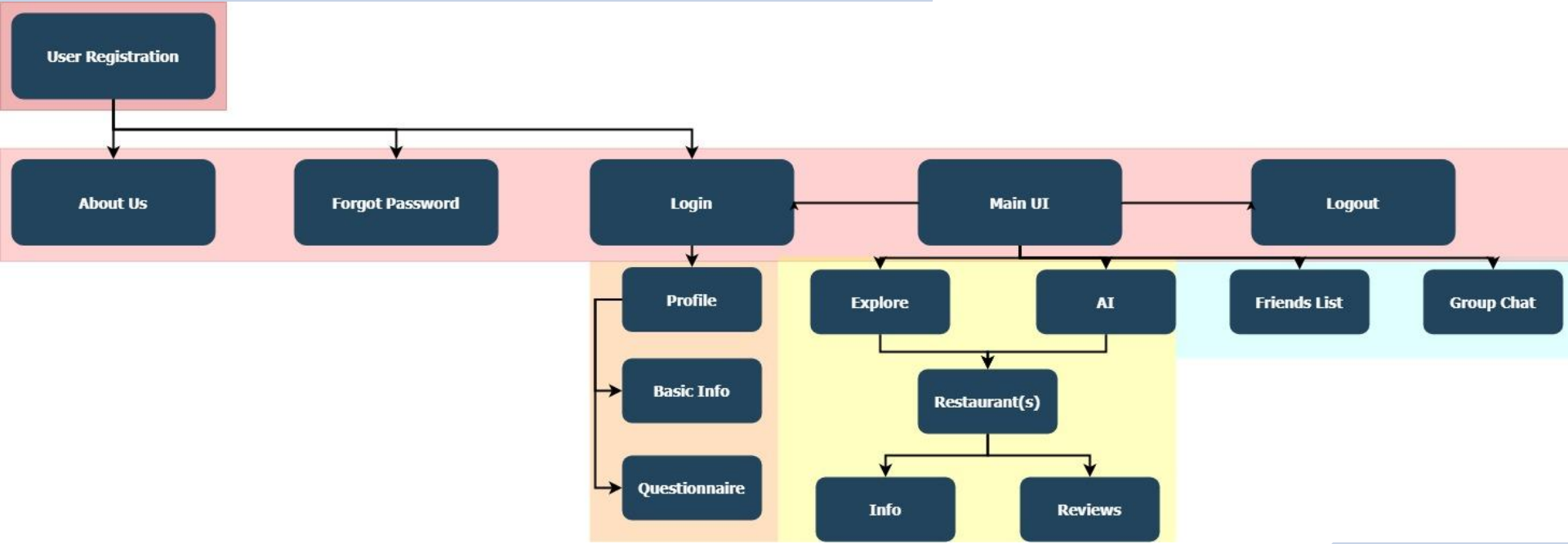
User Stories

#	User Story	Description	Priority	Notes
1	User login	A user wants to be able to log into the application (securely)	Must Have	Must be secure and information must be safely held somewhere.
2	Log out	A user wants to be able to log out of the application	Must Have	
3	Main menu	There must be a main menu to navigate the application.	Must Have	Should be intuitive.
4	User profile questionnaire	A user must be able to answer questions to set preferences for their profile.	Must Have	Asks main questions like top food types, price range, location. (maybe?)
5	Feedback for Quikpik	A user must be able to provide feedback for Quikpik developers to read.	Should Have	Linked to some resource we can read. Database?
6	User restaurant review	A user should be able to provide a review for a restaurant through Yelp.	Should Have	Linked to Yelp.

User Stories (Cont.)

#	User Story	Description	Priority	Notes
7	Friends list	A user should be able to add friends and view a friends list.	Depends	Might be best to save this for later iterations of application.
8	Group chat	A user should be able to create and engage in group chats with friends on their friends list.	Depends	Might be best to save this for later iterations of application.
9	Recommendation AI	The application must be able to make recommendations based on a user's preferences as listed on their profile.	Must Have	Extremely important for basic functionality of application.

Sitemap

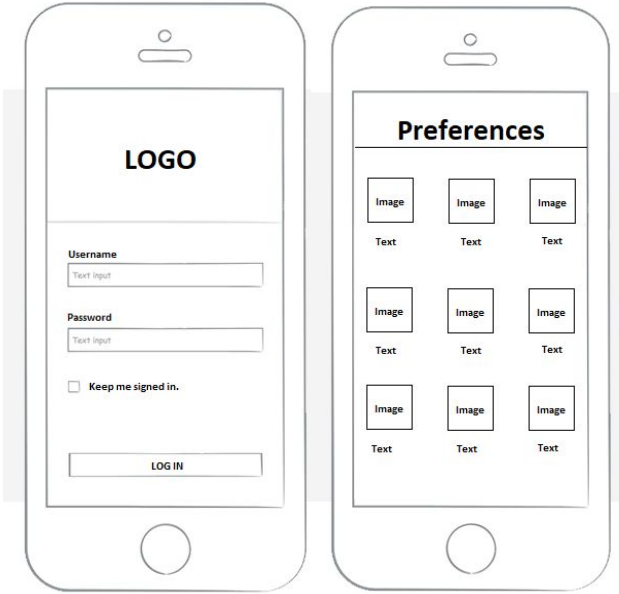

User Registration
About Us
Forgot Password
Login
Main UI
Logout
Profile
Basic Info
Questionnaire
Explore
AI
Friends List
Group Chat
Restaurant(s)
Info
Reviews

Page Descriptions

Page	Elements
Homepage	1. QuikPik Logo/Background 2. Sign-in/Sign-up Option
Sign-up Page	1. User Information 2. Email and Password 3. Security Questions 4. Create User Profile
Login Page	1. Email/Password 2. Forgot Credential Link 3. Feedback Link
Interaction Page	1. Chatbot Option(Change in user preference) 2. Search for restaurants
My Account Page	1. Change password 2. Change chatbot voice?

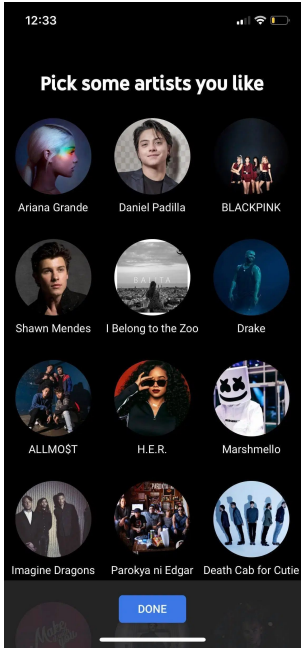
Wireframe of UI

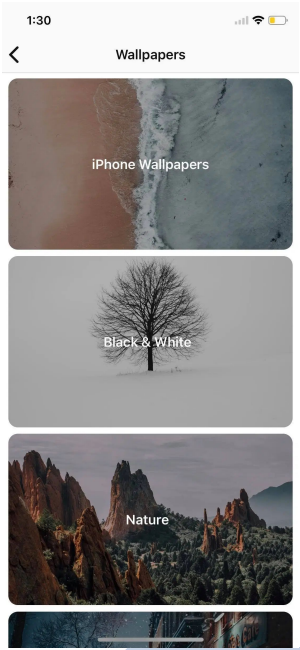
- Simple UI design that is descriptive enough to where users can know how to navigate through the interface but simple enough to where it's not stressful to do it
- The main focus of this type of user interface is allowing for users to move through to get what they need quickly without hassle



Interface

- Although there is a priority on simple design, we want to achieve a good aesthetic that users will like
- Using vibrant colors that stand out, properly spacing everything out, and making sure everything is easy to see





Additional Functional Requirements

- **Data Security:** The user's data, which includes their login information, initial questionnaire preferences, and security questions, must be secure against harmful entities.
- **Version Verification:** The current version of the app must be identifiable on android devices to be able to update it with newer releases when necessary.

Non-Functional Requirements

- **Usability:** The user should be able to learn the app's functionalities, such as logging in, answering the questionnaire, and traversing the main menu, with minimal difficulty.
- **Security:** The user's information, such as their login information and preferences, should be protected from unauthorized access and viruses.
- **Performance:** The app should perform smoothly so that the user's experience isn't hampered by slow loading time.
- **Reliability:** The app should function without crashing and contain minimal software bugs.

Performance Requirements

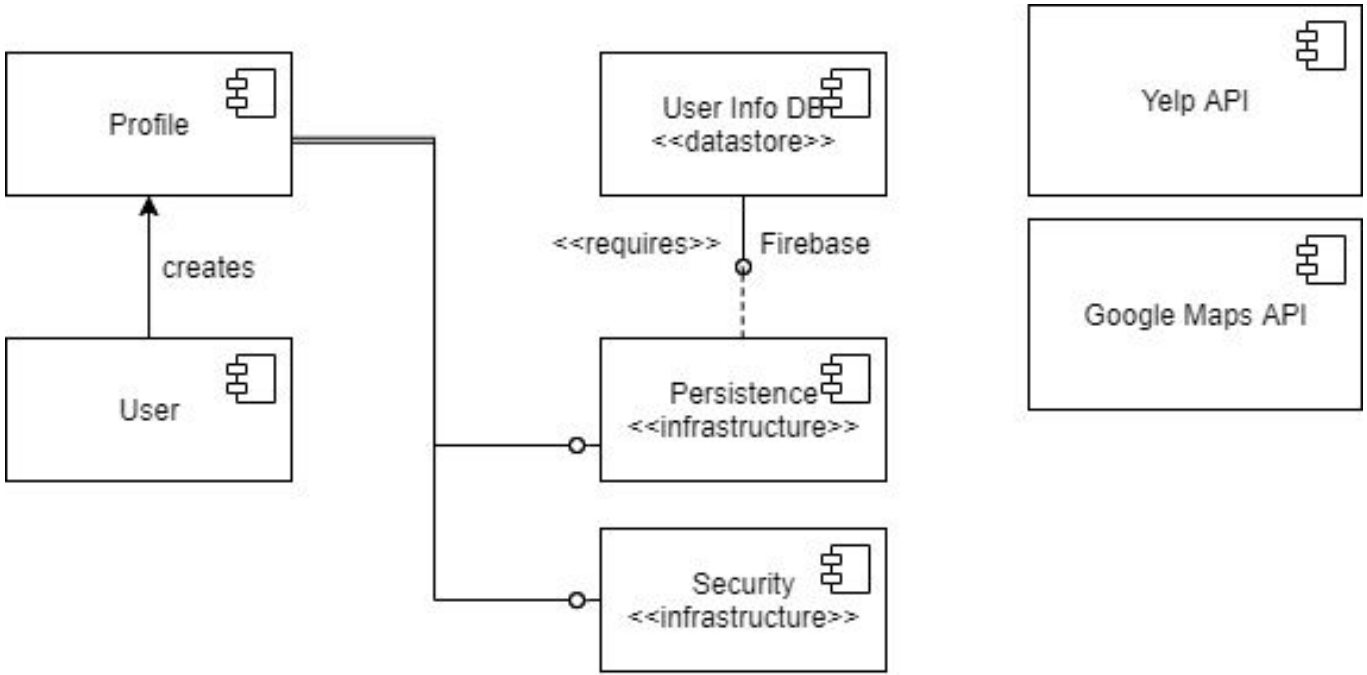
- Start-up time should take 1-2 seconds to load.
- App should not be consuming a high amount of battery life in order not to cause the device to heat up.
- App should use less than 15% of the memory at all times.
- App needs to run smoothly on various RAM and memory processors.
- App should remain in the same state if it is removed from focus and revisited.
- App should run smoothly on various network speeds such as Wi-Fi and mobile networks, 3G and 4G when switching between different networks.
- App should work on various Android devices.

Future Iterations?

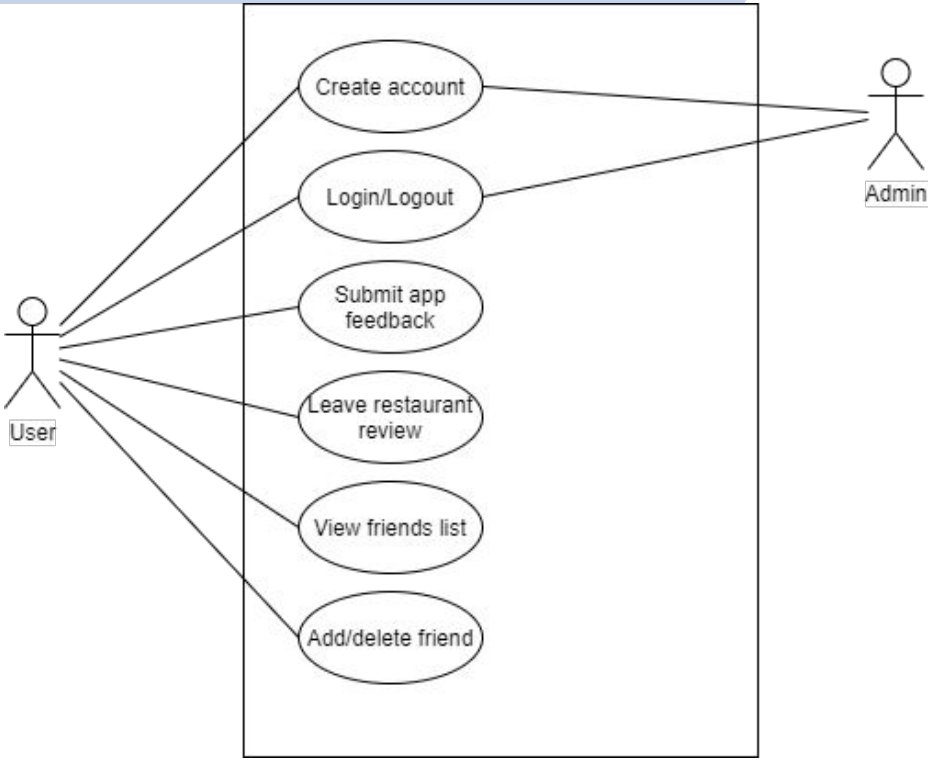
For future iterations of the Product Requirements, we will add more User Stories and Non-Functional Requirements as we go forward on our project. When we achieve a working demo, we can better define our Performance Requirements, Wireframe and Interface.

Architecture and Design Document

System Component Diagram



Use Case Diagram



Tradeoff Analysis

Database			
Criteria	**Firebase**	**MongoDB**	**PostgreSQL**
Team Familiarity	1	0	0
Time Constraint	1	-1	-1
Maintainability	0	1	1
Flexibility	1	1	-1
Scalability	1	2	1
Authentication	1	1	0
Security	1	2	0
Performance	1	2	1

Tradeoff Analysis cont.

Cost	-1	-1	1
Data Synchronization	1	1	1
APIs	1	1	0
Supported Languages	1	1	1
Learning Curve	1	-1	1
Sum +'s	11	14	7
Sum 0's	2	1	4
Sum -'s	1	3	2
Net Score	10	11	5
Rank	2	1	3

Risk Management (1 of 3)

ID	Description	Mitigation Scheme	Severity Level	Date of Identification	Status
R1	User information database falls victim to SQL injection attack and we lose user data.	Using parameterized statements whenever possible when creating login and password in database	High	02/27/20	Ongoing
R2	User password not encrypted and visible in database	Using hash functions or some sort of encryption to lock away user passwords	High	02/27/20	Ongoing
R3	Inadequate machine learning model for chatbot	Constant research on classical machine learning models	High	03/01/20	Ongoing
R4	Chatbot doesn't recognize speech	Think about a Semantic analysis based model. We always revert to secondary design of text based bot.	Med	03/01/20	Ongoing

Risk Management (2 of 3)

R5	Chatbot isn't conversational	Constantly researching a machine learning model that uses user chat, profile, and yelp review data as a corpus for ML model. We can always revert to secondary design of text based bot.	**Med-High**	03/01/20	Ongoing
R6	User suggestion isn't personalized	Implement the Yelp API so we have a backup suggestion style if ours fail	**Med**	03/01/20	Ongoing
R7	Activity monitoring and data retrieval issues	Creating a centralized database that is efficient in storing and retrieving user information.	**High**	03/03/20	Ongoing
R8	Version of user device is outdated to run the app	This app can be run on 99.8% of Android devices. Android Studio has this option to control which devices can run the project.	**Low**	03/03/20	Incomplete

Risk Management (3 of 3)

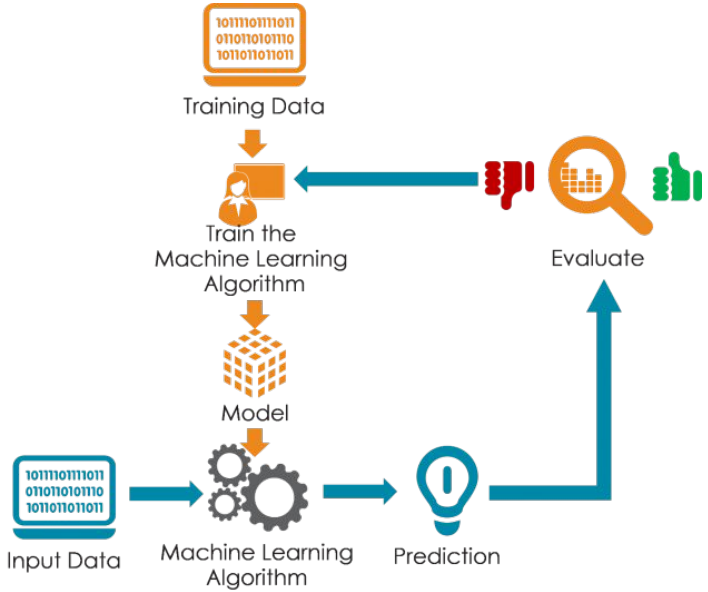
R9	Users won't like some aspects of the app	Implement a feedback hub that the customer can write their concerns on.	Med	03/04/20	Incomplete
R10	Slowed development from group because of other classes	Assign coding goals and assignments in pairs. So if one member is busy the other can still work	Med	03/07/20	Complete
R11	Performance on the app is too slow	Optimize device performance by monitoring the back-end effectively	Low-Med	03/07/20	Incomplete

Structure of our code so far

- User platforms planned
  - Android
- Coding language
  - Java and Python
- Types of databases planned
  - Firebase
- 2 APIs and 1 Server
  - Google Maps API and Yelp API and Firebase

Machine Learning

- Different from predictive modeling
- Machine learning model is data driven while predictive modeling is use case driven
- Plan on deploying our machine learning model to handle suggestions based off user preferences and history

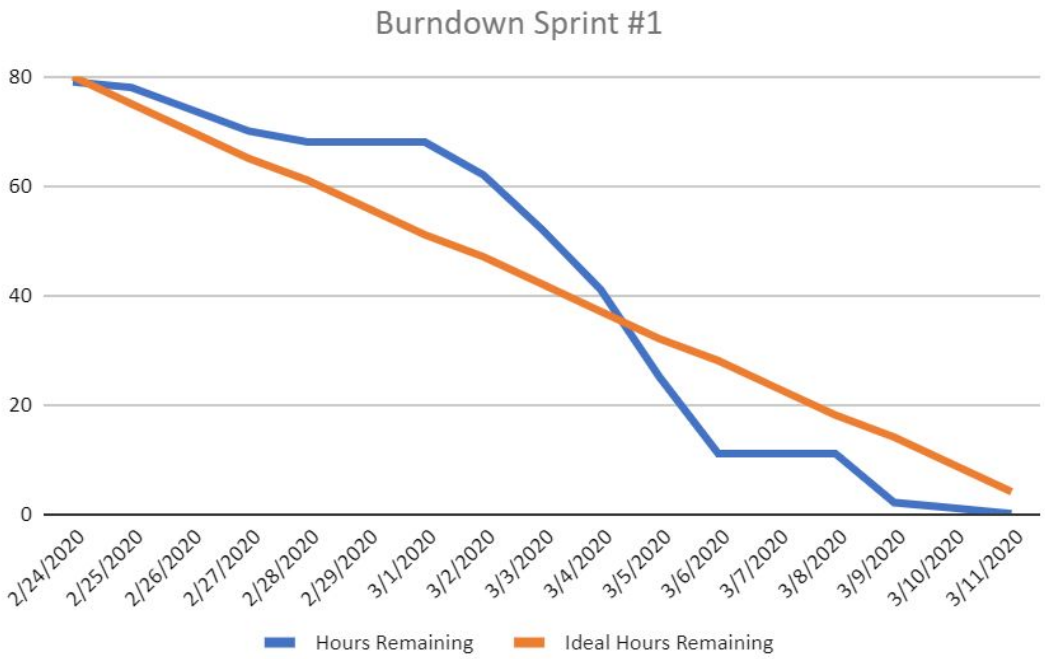


Sprint #1


Boards
Trello
Quikpik Project 2020
Quikpik Dev Free
Private
JV AP EA +2
Invite
on Elegantt »
Activate your PRO trial
Burndown Chart
Butler
Show Menu
Product Backlog
Main Menu
User Profile Questionnaire
Feedback for Quikpik(User --> Developers)
User Restaurant Review (Linked to Yelp)
Make Recommendation AI
Develop AI to be based off of user preferences, habits, location
Develop friend lists, group chats for users
Add another card
Sprint 1 Backlog
Set Up Environment
User Login (Secure and Safe Information)
Mar 9
AP EA
Log out option
Mar 9
AP EA
User Basic Interface of App
Mar 9
AP EA
Link Yelp API to App
Mar 9
JV
Link Google Map API to App
Mar 9
R
Product Requirement (Goals)
Mar 9
JV
Product Requirements (User Personas)
Mar 9
R
Product Requirements (User Stories)
Mar 9
JV
Add another card
Documents
Product Requirements Document
Mar 9
1
Architecture and Design Document
Mar 9
1
Add another card
To-do
Add a card
In Progress
Add a card
Testing
Add a card
Completed Sprint #1
Add a card
Add an
SPRINT BOARD (Feb. 24, 2020)
17 Story Points planned

SPRINT BOARD (Mar. 9, 2020)
15 Story Points completed

Burndown Chart



Sprint Retrospective

Did the team meet its Sprint Goal?

Our goal was to finish the PRD, ADD, and work on implementing the APIs. We finished the documents but not all the code.

3 In Progress user stories

What was the team's velocity?

Total velocity = Hours Per Week * Amount of Weeks in Sprint * Amount of Team Members

= 5 hours per week * 3 weeks * 6 Team Members = 90

Sprint Review - Held on March 9, 2020

Plans for next Sprint

- Training the Machine Learning Model
- Combine our APIs with our server