Architecture and Design Document: Quikpik

California State University, Long Beach

CECS 491A-Sec 11 Spring 2020

March 16th, 2020

Ernie Argel 017984237
Jalon Flores 015843540
Ranjit John 016695989
Anthony Pham 014415919
Judy Tran 017194591
Juan Villa 015909255

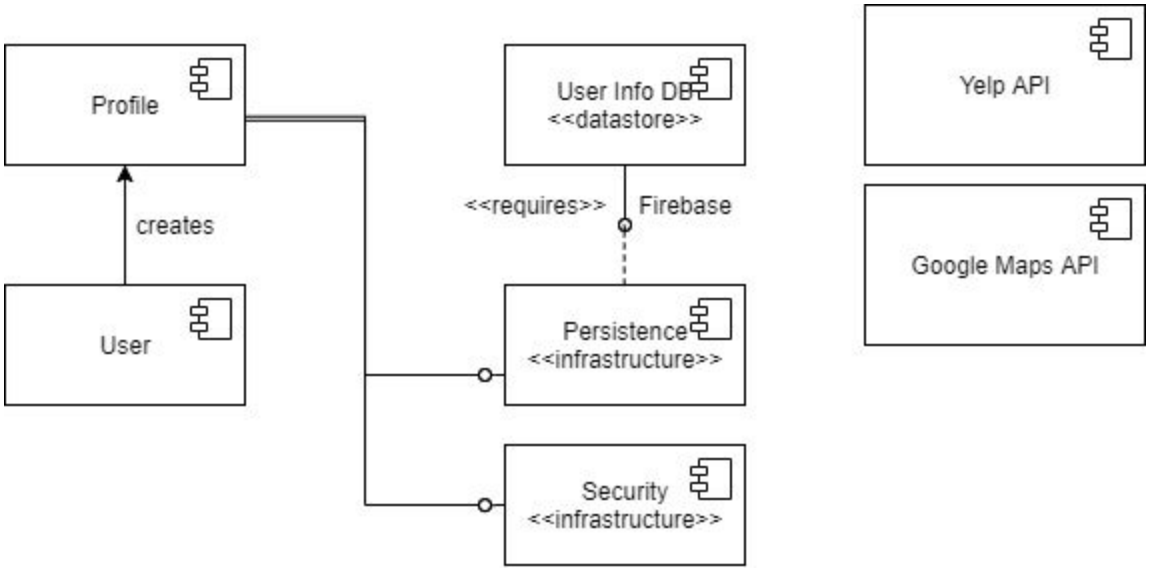
Table of Contents

Version History

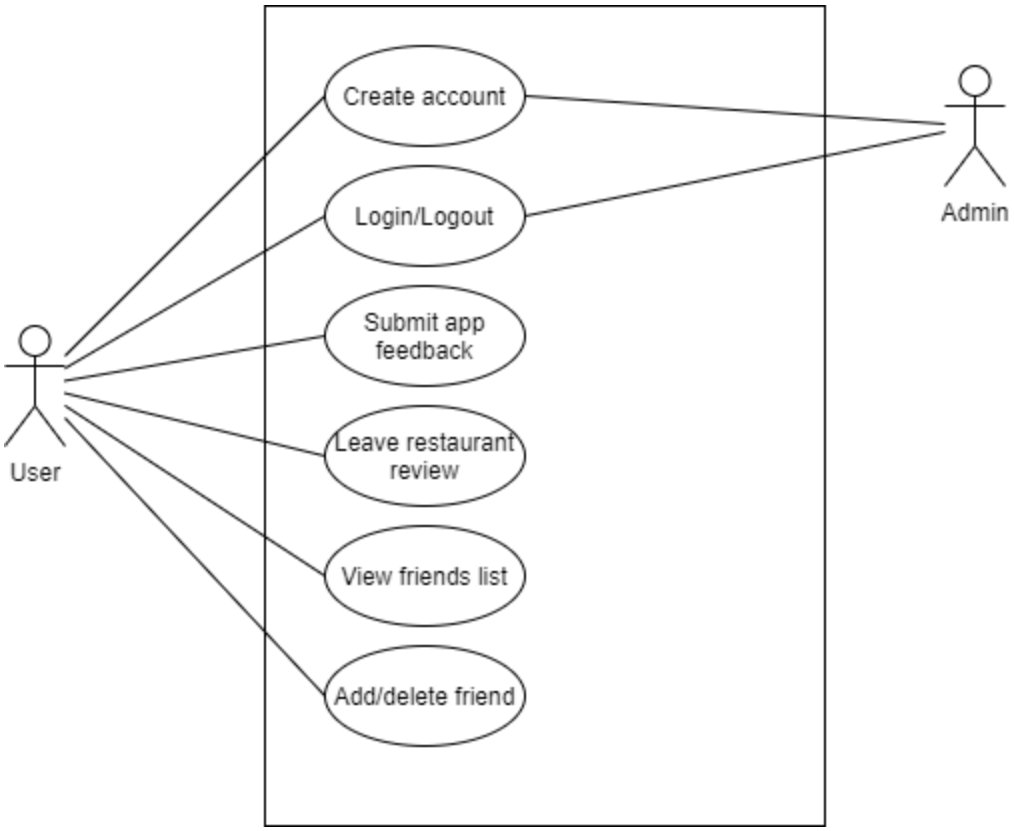
Version No.	Date
1.0	March 16, 2020

Diagrams

System Component Diagram



Use Case Diagram



Tradeoff Analysis

Frameworks			
Criteria	**Android Studio**	**React**	**Angular**
Team Familiarity	2	0	0
Time Constraint	1	0	-1
Scalability	-1	2	1
Maintainability	1	2	1
Performance	-1	1	2
Portability	0	1	1
Testing	0	-1	1
Flexibility	1	1	0
Learning Curve	1	-1	-2
Sum +'s	6	8	6
Sum 0's	2	2	2
Sum -'s	2	2	3
Net Score	4	6	3
Rank	2	1	3

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
Cost	-1	1	-1
Data Synchronization	1	1	1
APIs	1	1	0
Supported Languages	0	2	1
Learning Curve	1	-2	1
Sum +’s	10	14	6
Sum 0’s	2	1	4
Sum -’s	1	3	3
Net Score	9	11	3
Rank	2	1	3

Risk Management Table

About:
Our group has identified some possible risks that might affect us in the development of our app. The table below is divided into six sections: id(identity number of the risk), description(What the risk is), mitigation scheme(plan to combat/avoid the risk), security level, date identified and status.

Severity Levels:
High - high impact on the app if risk occurs.
Medium - moderate impact on the app if risk occurs.
Low - little or no impact on the app if risk occurs.

Status Level:
Ongoing - Members are currently addressing the risk.
Complete - Members addressed the risk and took necessary steps to mitigate the risk.
Incomplete - Members have not addressed the risk.

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

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
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