

EATIFY

Most delicious Android app, it's worth to try :)

150114002 - Ayşenur Yılmaz

150109009 - Ceren Birinci

524118023 - Mustafa Zortul

1.0 Introduction

This section of this document covers purpose and scope of the Eatify application. Also, overview of the requirement specifications can be found in this section.

1.1 Purpose

The purpose of this document is to define and describe the requirement specifications for Eatify.

1.2 Scope

Eatify is an android application that will provide a platform to learn and share recipes with others.

One of the goals of application is to provide users an elegant interface, so they can extract valuable information that needed in order to understand recipe with minimum waste of time.

With its dynamic design, application interface will allow different types users to use the platform as they desire.

1.3 Definitions, Acronyms and Abbreviations

Definitions: None

Acronyms: SHA256 : Secure Hash Algorithm

IDE : Integrated Development Environment

Abbreviations: None

2.0 General Constraints

Software limitations:

- System shall need 7.0 or a higher version of Android Operating System.

- Application will be developed with Android Studio 3.3.1

Hardware limitations:

- Each user shall need a Android 7.0 or higher version compatible device. The database shall be stored on an Android SQLite Database for caching purposes and lowering content access duration.

3.0 Assumptions and Dependencies

- Each user must create their own profile.
- Users must log in to the system with their valid user id and password which they created before.
- Server must be running for the system to function without major interrupts.
- Only the Administrator can delete user informations.

4.0 Requirements

This section of document covers functional and non-functional requirements of Eatify.

4.1 Functional Requirements:

4.1.1 User Login

In order to view applications content user must login with their login credentials which are username and passwords.

If users device doesn't have any cached credential data or cached credential data is no longer valid user should be asked to login.

If users input matches with a account credential that stored on server side, an information pop-up must inform user about successful login attempt,entered account credentials should be cached on device for further logins and view should switch to main view of application.

If users input doesn't match with any account credential stored on server side, an

information pop-up must inform user about successful login attempt.

4.1.2 Search

User should be able to search recipes by words.

User should be able to filter search results according to recipe attributes.

If internet connection error occurs, user should be informed.

4.1.3 Categorization

After selecting one or more category, user should be able to filter recipes according to options below.

- Past 24h/week/month/year/all time top recipes
- Last uploaded

If internet connection error occurs user should be informed.

4.1.4 Viewing recipe

Once user clicks on a recipe, user should see information related to the recipe.

4.1.5 Rating

User should be able to rate recipes via upvote, downvote buttons.

4.1.6 Comments

User should be able to comment below recipes or other comments, once comment made recipe poster should get notified according to application preferences.

4.1.7 Recipe Submission

- Users will write detailed information about recipes, like which ingredient will be used first, which will be mixed after etc.
- Ingredients will be explained by owners clearly with their amounts.
- There will be information about recipe for number of servings.
- Also users will be able to upload a photo of related recipe.

4.1.8 Following

User should be able to follow each other with follow button on other profiles. If a user gets followed by another user, according to application preferences followee should get notified.

According to application preferences user should get notified if one of the followers uploads a recipe.

4.1.9 Preferences

- Notification settings
- Account settings
- Interface related settings, disable or change some features

4.1.10 Report

User should be able to report comments, recipes and profiles, if he or she thinks features getting used outside of their intended use.

Reported comments and recipes should be sent to moderators, so it can get reviewed and decided about its removal by administrator.

4.1.11 Bookmark

Users will be able to bookmark recipes, for ease of later access.

4.2 Non-Functional Requirements

- Availability: Users must be able to reach their accounts 24 hours a day, 7 days a week with 90% uptime.

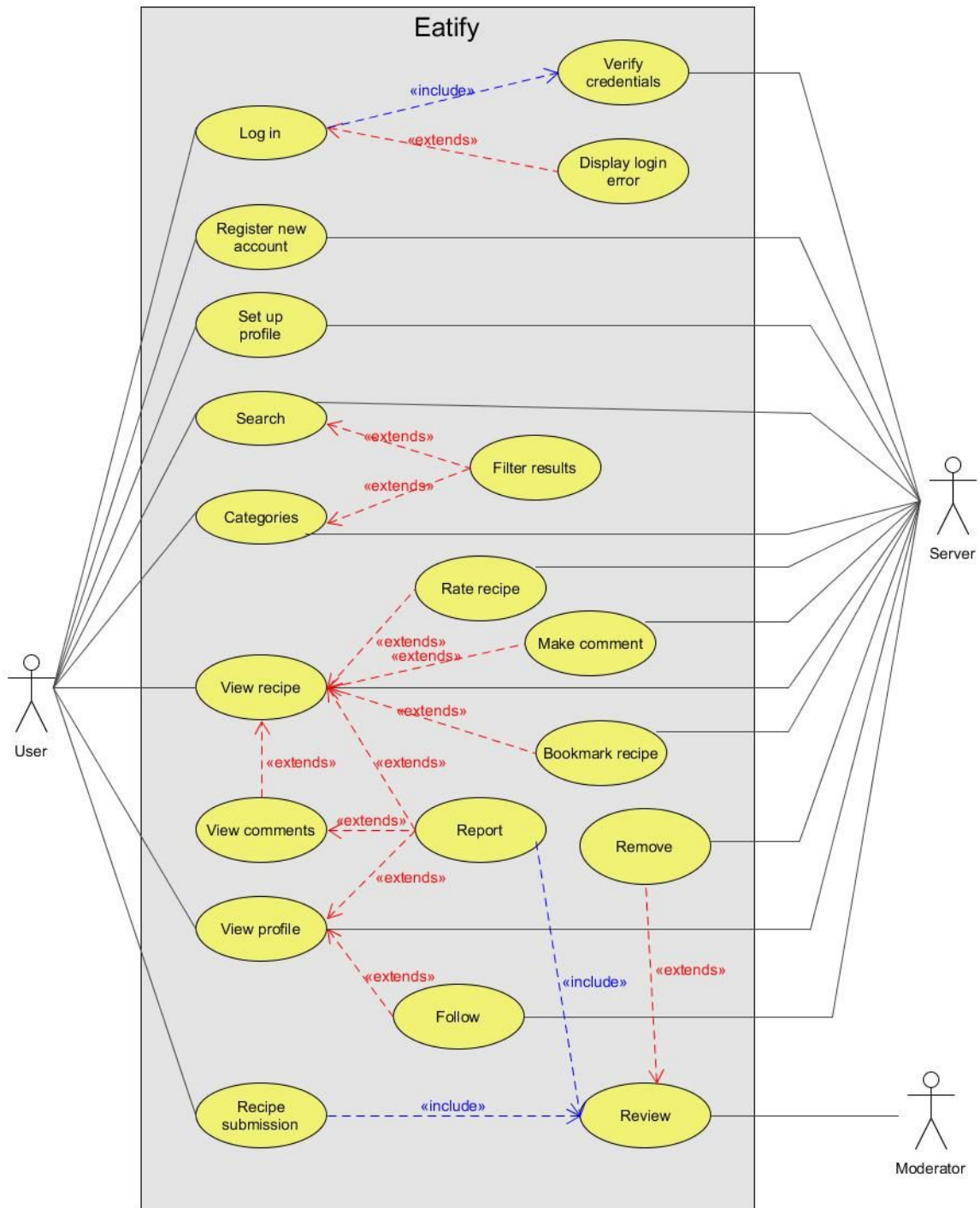
-Performance: Performance should be working on mobile devices; most operations should be done in at most 3 seconds in the last 4 year's hardware.

-Reusability: All software that runs on a user device shall be written in a prevalent programming language such that the software can be run on a personal device without having to download a supporting environment.

-Usability: The user interface should be easy to understand.

-Security: Username and password strings entered by user should be concatenated and hashed with SHA256 on device, then username, hash value pair should be sent to the server to be stored.

6. Use Cases



Who did what:

Ayşenur Yılmaz	Use-case diagram, general constraints, assumptions and definitions portions.
Ceren Birinci	Non- functional requirements specification, submission, assumptions and definitions portion.
Mustafa Zortul	Use-case diagram,functional requirements specification.

In general, we have to say that all documents those submitted by us prepared all together, so it is kind of hard to say who did what actually.

For every submission we had two or three times meetings and that's why we are all wrote these documents.