



Capstone Project Demo 1

System Requirements and Design Document

by Brute Force



Authors:

Mia Gerber
Matthew Perry
Wanrick Willemse
Duart Breedt
Linda Potgieter

Contents

| | | |
|----------|---|----------|
| 1 | Introduction | 2 |
| 1.1 | Purpose | 2 |
| 1.2 | Scope | 2 |
| 1.3 | Overview | 2 |
| 2 | Overall Description | 2 |
| 2.1 | System Interfaces | 2 |
| 2.2 | User Interfaces | 3 |
| 2.3 | Hardware Interfaces | 4 |
| 2.4 | Software Interfaces | 5 |
| 2.5 | Product Functions | 6 |
| 2.6 | User Characteristics | 6 |
| 2.7 | Assumptions and Dependencies | 7 |
| 3 | Specific Requirements | 8 |
| 3.1 | External Interface Requirements | 8 |
| 3.2 | Hardware Interface | 8 |
| 3.3 | Software Interface | 8 |
| 3.4 | Communications Interface | 8 |
| 3.5 | User Interface | 8 |
| 3.6 | Functional Requirements | 8 |
| 3.7 | Performance Requirements | 11 |
| 3.8 | Design Constraints | 11 |
| 3.9 | Software System Attributes | 11 |

1 Introduction

1.1 Purpose

Application for splitting a restaurant bill between friends. The application will be used as a utility in a social context to decrease the time it takes to work out how much everyone needs to contribute to a shared bill.

1.2 Scope

This project will essentially be divided into two phases, the first of which focuses heavily on user experience and the second accuracy.

Phase One: A multi-platform mobile application that users can send a photo of a bill to and receive an interactive interface generated from the data contained within that photo that they can then use to work out correct totals.

Phase Two: An artificial intelligence agent that has been trained to recognise text in a photo in a very specific way will be the backend mechanism that can be given any photo and will return a JSON object with the structure of that photo that can be parsed to create the interface in phase one.

1.3 Overview

The goal of this document is to define a clear project scope that will be used to contain and guide coming deliverables as well as give clarity as to how our application will function as a system built up out of well-defined components.

2 Overall Description

2.1 System Interfaces

The Split Bill system has no existing external interfaces. The system will consist of a webbased OCR system that detects and identifies a receipt format. The webbased system interfaces with a user's mobile device.

2.2 User Interfaces

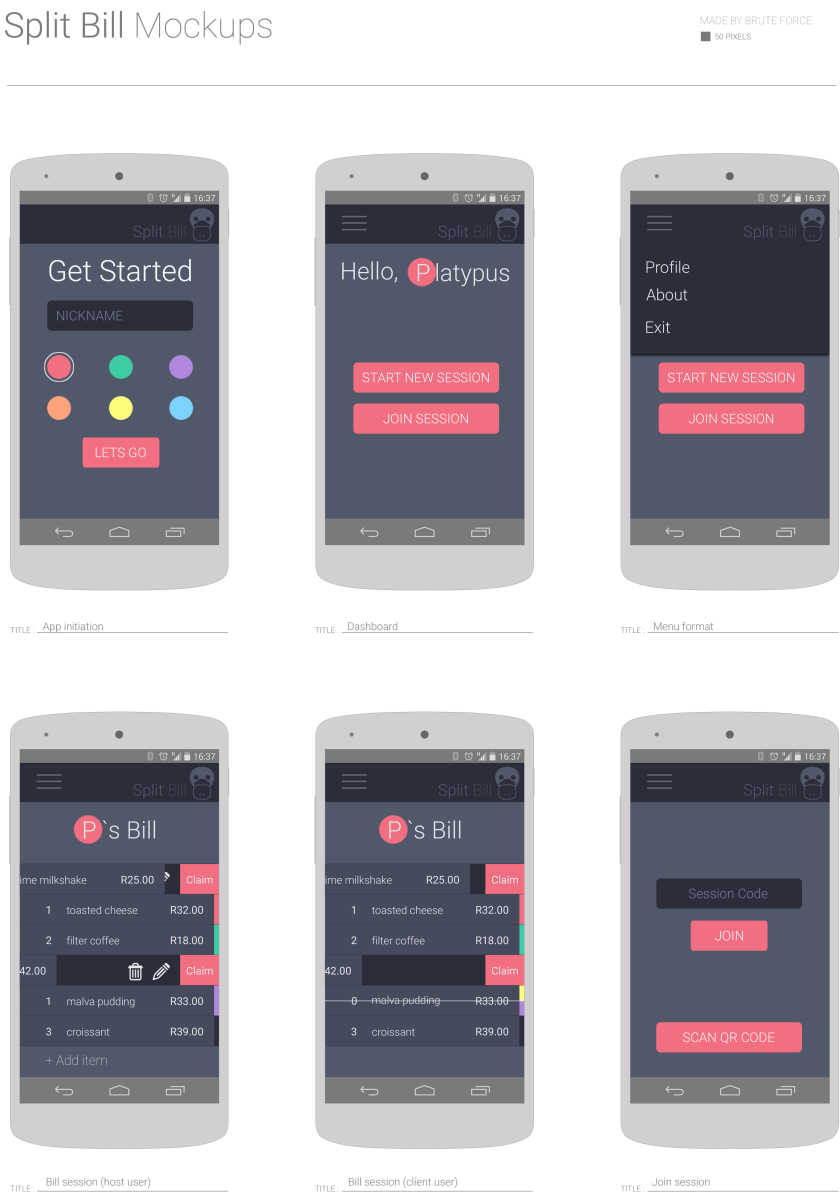


Figure 1: Interface Mockup

2.3 Hardware Interfaces

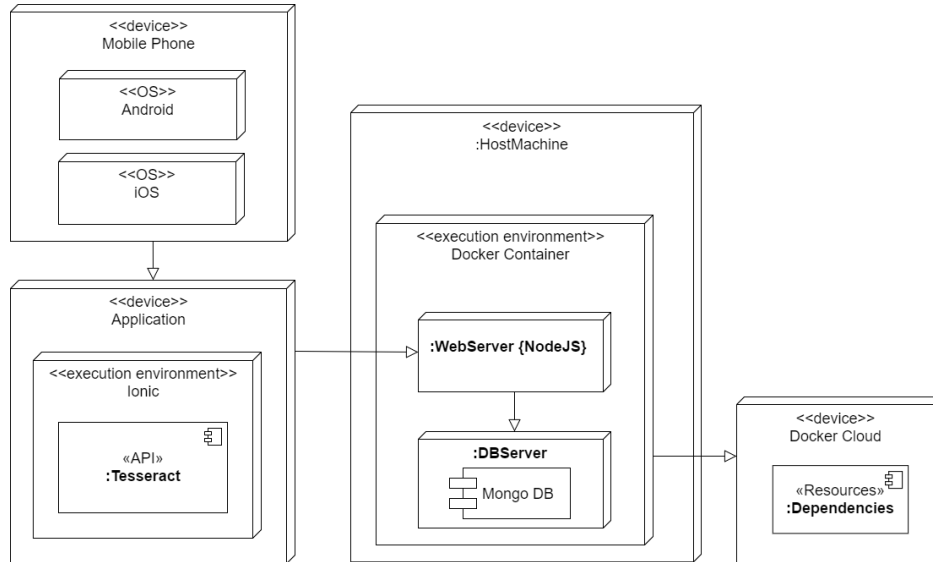


Figure 2: Deployment Diagram

The Split Bill application will make use of the camera on the device. The data modules (WiFi and mobile data) will be used for connection to the server over the internet.

2.4 Software Interfaces

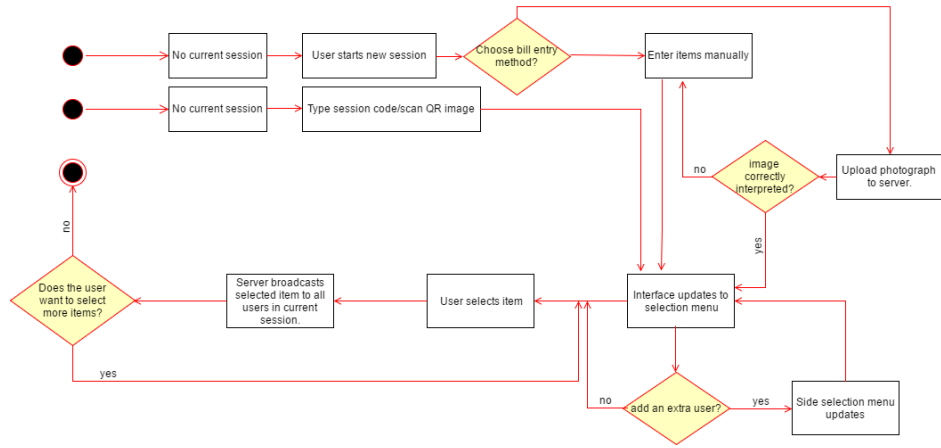


Figure 3: Activity Diagram

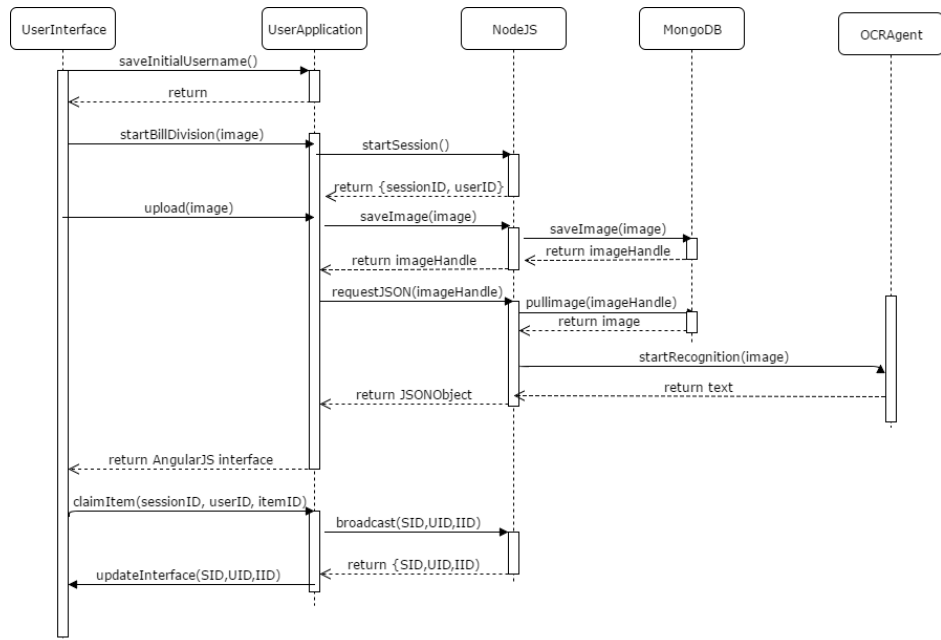


Figure 4: Sequence Diagram

2.5 Product Functions

Using the mobile interface, a user will be able to take a photo of a restaurant receipt. The mobile device will position and crop the image to only include the restaurant name, items and amounts listed and the totals. On successful upload a user session is created. The session ID is returned to the initiating user's mobile device. The mobile app then provides a session ID in text format that can be shared with other users using a QR barcode or NFC.

The image is uploaded to a webserver where OCR is used to extract the information on the receipt. The information extracted from the image will be analyzed and an AI will determine the closest matching receipt format. The information from the image along with the determined format will be used to create a JSON object that is returned to all the users connected to the session.

Each user will be able to set a nickname and avatar colour to identify each user. A user will be able to select items from the receipt that is then regarded as claimed, blocking the item for every other user in the session. As a user claims an item, his total will be updated. Once all the items have been claimed, a user can choose a tip percentage. The user's total contribution will be displayed to the user.

2.6 User Characteristics

Our application is being marketed as a "utility" and as such we won't be targeting a specific user, we want to appeal to as many people as possible because the type of person who visits a restaurant with friends is almost every person imaginable. With that said we will however assume our users to be literate in mobile technologies. A basic understanding of touch screen gestures and navigation between different screens will be assumed. We are making HCI (Human Computer Interaction) a major focus during development of our GUI and as such the aim is simplicity and usability without losing aesthetic appeal.

| Actor: User | System: Application |
|---|---|
| | 0: System displays "Create" or "Join" menu options. |
| 1: The user opts to create a session | 2: The system instructs the user to take a photo of the bill and activates the camera |
| 3: The user takes an image | 4: The system displays the extracted items and total |
| 5: The user selects items from the bill | 6: The system updates the user's total |
| 7: The user confirms selection | 8: The system prompts for a tip percentage |
| 9: The user enters a tip percentage | 10: The system displays the user total and a breakdown |

Figure 5: Actor-System Diagram 1

| Actor: User | System: Application |
|--|---|
| | 0: System displays "Create" or "Join" menu options. |
| 1: The user opts to create a session | 2: The system instructs the user to take a photo of the bill and activates the camera |
| 3: The user takes an image | 4: The system displays the extracted items and total (Options to add user locally or via session is visible) |
| 5: User chooses to add another user locally | 6: The system prompts user nickname and avatar colour |
| 7: User enters information and confirms | 8: System displays the bill items and total |
| 9: The user selects items from the bill for local users simultaneously | 10: The system updates the user's total |
| 11: The user confirms selection | 12: The system prompts for a tip percentage |
| 13: The user enters a tip percentage | 14: The system displays the user total and a breakdown |

Figure 6: Actor-System Diagram 2

2.7 Assumptions and Dependencies

- All team members are available and have the necessary skills and knowledge to collaborate and complete the project. Set deadlines and milestones are achievable and project can be finished by the October deadline.
- There will be an external server provided to which data can be sent from applications that have Internet connection.

- The application will be open source. This means open source libraries will be used for most of the applications components such as including more complex aspects such as OCR and peer to peer connectivity.
- OCR technology will be able to convert images uploaded to the server to a standardized file format.
- The application will have to be able to operate on all available mobile operating systems.

3 Specific Requirements

3.1 External Interface Requirements

3.2 Hardware Interface

The Split Bill system will be designed to be run off of a smart phone. The system will make use of the smart phone's camera in order to take a picture of the bill for processing.

3.3 Software Interface

The user will interact with the application through a touch screen interface on the device.

3.4 Communications Interface

The WiFi module, or the mobile network module will be used to establish a connection over the internet to the central server. Through this server, the user will be able to communicate with other users and split the bill with them.

3.5 User Interface

The user interface will be a full screen application that will be designed following HCI protocols to ensure that the optimal user interface that will serve all the functional requirements.

3.6 Functional Requirements

1. The system must be able to recognise text on a bill.

2. The system must be able to correctly categorise the different "columns" of text into discrete structures that represent different information semantically (e.g. an item of food is not interpreted as a numeric amount)
3. The system should consciously check whether it has correctly interpreted the bill and if it did not ask the user either for a higher quality photo or as a last resort give an error message.
4. The system should return all information gleaned from the OCR process as a JSON object of which the structure is known to the component that will be receiving the object.
5. The user interface provides a facility for both taking a photo and sending it to the server.
6. The user interface provides a facility for starting a new session which will return a temporary session code which is shared with users who also have the application around the table.
7. The user interface which is generated after interpretation of the bill (after the associated JSON object has been received) contains a facility for choosing items to pay for and generates a running total as the user decides.
8. All user that were added to the current session will receive the exact same interface and as such will be able to see which items are still available to be paid for and which are no longer available. (Asynchronous calls to the server will facilitate updating of the user interface)
9. Additionally there must be a facility for users to manually create a bill by entering items and their price into a form. (the final interface will then be created from the information entered here) This will also be used as a fallback if OCR fails.
10. Users must be uniquely identified by name so that everyone around the table is able to see who is taking responsibility for which item. When the app is opened initially the user will enter a name and choose a color that will be the background color of their icon during an active session.

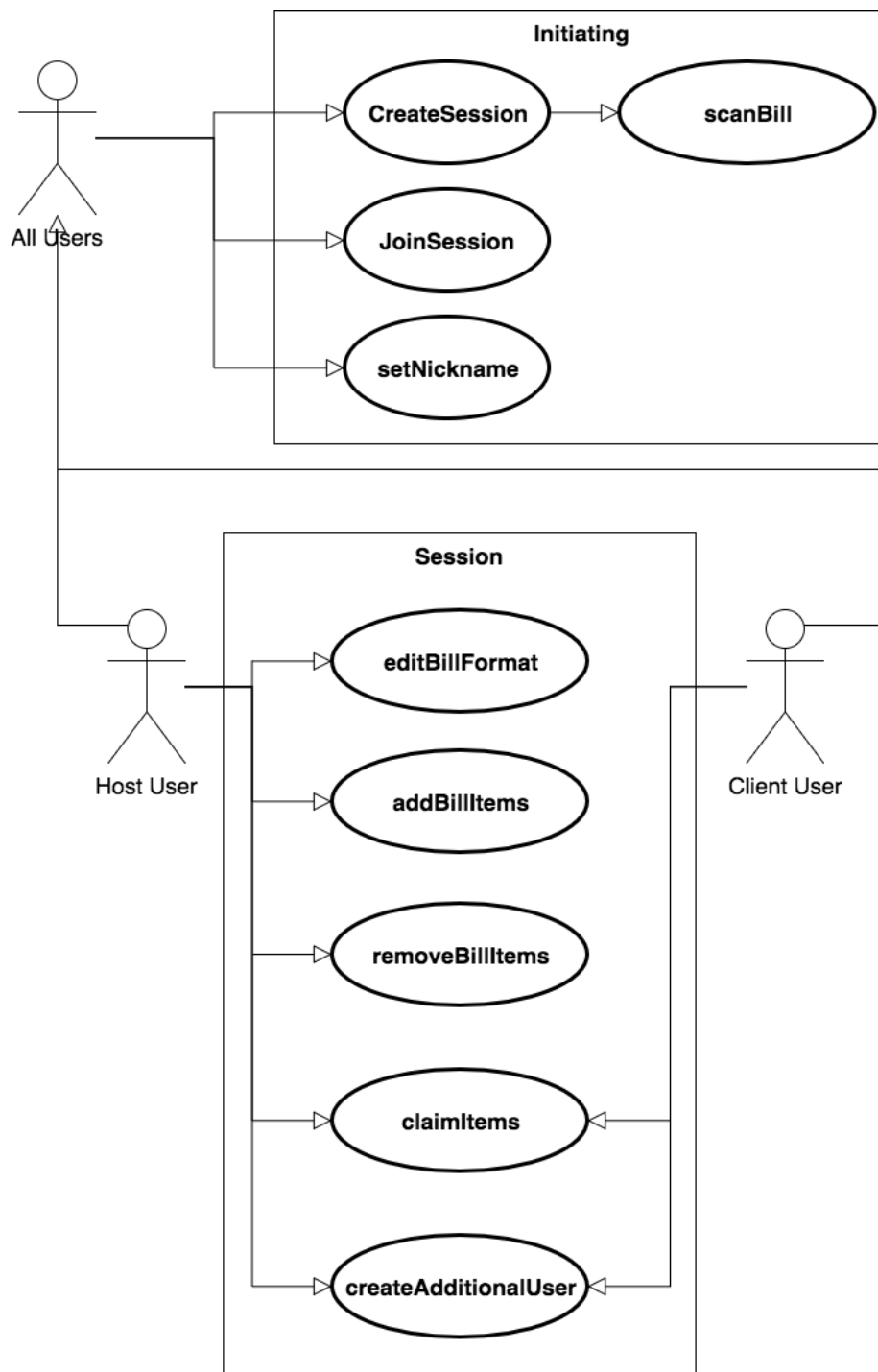


Figure 7: Use Case Diagram

3.7 Performance Requirements

Since the application is user based, performance is of utmost importance. The application should be very responsive and be able to process user information and requests in an appropriate amount of time to ensure the user has a pleasant experience. Performance requirements will focus on response time, completion time, efficiency and speed up, without compromising on functionality and effectiveness.

3.8 Design Constraints

- Even though we will not be persisting any image data for longer than necessary the size of images being sent to the server is a constraint we will need to be conscious of, possible solutions will be cropping or simply reducing image quality by changing the image to only use black and white or a degree of grayscaling.
- The experience of the development team might influence the degree to which the artificial intelligence portion of this project comes to fruition.
- The possibility of getting the application to still be functional without internet access can become a design constraint later in development.

3.9 Software System Attributes

The app will be designed for off-line use as well as for when it's connected to the server. Thus the app will always be available to be used. The system will not store any persisted user data at any point so security will not be an issue. Making use of Docker for deployment of the system, that will ensure that there will be a 99% uptime. The application will communicate with the server to get data about how to interpret the bill, and send learnt data to the server to keep other users up to date. Communications with other users and their devices will happen via the server, so that the bill may be split across multiple devices.