

# Non-Functional Requirements Specification

## Beverage Booker

### **System-Wide Functional Requirements**

- Auditing: It is a need to track who used the system. This will allow for tracking of orders to allow orders to be attributed to the proper owners. Also it is crucial to always track transactions.
- Authentication: Access to the system will be controlled as to disallow access to those who do not need to access specific parts of the system while allowing those who need access to do so. This will be done through levels of access in which an administrator account will exist to allow those that need to access specific areas of the system to do so.
- Scheduling: System actions such as maintenance and other things should be scheduled on the system e.g. have the system shutdown at a given time to allow someone to access it.
- Security: Elements of the system such as payment methods and accounts will have to be secure. To do this the system will have to incorporate an encryption mechanism to keep private information safe from sources such as hackers.

### **System Qualities**

- Usability
  - Accessibility: The application should be accessible from remote locations. It is available to them at any point. It is going to be available on both browser and mobile phone (probably android).
  - Ease of use: A user should be able to access the system and be able to place an order easily. This can be done by making the system's UI and other components simplistic to allow for ease of use. They should easily create an account or if they choose not to, not be impeded by the login screens.
  - Ease of learning: A user should easily be able to learn how the system operates and shouldn't have to put in a lot of effort to place an order. They will be able to learn easier with a simplistic and self-explained UI e.g. buttons that say what they do like "continue to checkout" or "continue to food items".
  - Task efficiency: A user should be able to use the system with as few mouse clicks/screen presses as possible, to do this the UI the user is presented with should be easily navigated with few buttons to allow for simplicity. There should

also be few pages the user has to go through to finalise their order to allow for minimal time usage.

- Ease of remembering: The system should be simple enough that a user is able to remember how to use it after using it once. Upon that user using it again they should be able to easily remember how to operate the system.
- Understandability: While the user is operating the system they should be able to understand the processes taken to accomplish their order. They should understand why they are being prompted to enter information and specify what they want to order. They should understand prompts of fields they failed to enter e.g. if they click continue to checkout without actually having anything in their cart then they are given a simple prompt stating they cannot continue as they have no items in their cart.

- Reliability

- Accuracy: Calculations performed by the system require 100% accuracy, as the system handles money and therefore it is of large importance that any calculation regarding money, providing the total of the order to the user, apply discounts, withdrawing that money from the user's account must be completely accurate for the user to have trust in the system.
- Availability: The system should be available all the time from 7 AM to 7 PM, maintenance access should be scheduled outside those hours specified before.
- Recoverability: The system should not be able to go down for more than an hour to prevent much impact to the business.
- Frequency of severity of failures: critical defects should not occur as long as backup files are often updated to current versions of the files, this prevents loss of large amounts of data and allows the system to be reverted to a previously functional state. if a defect occurs its expected the system can be put back to its previously functional point.

- Performance

- Response time:
  - Any interface the user operates should not exceed a response time of 2 seconds.
  - An account should be updated with the order it made within 3 minutes.
- Throughput: It is expected that the system is able to process many orders, transactions and updates at any given time.
- Capacity: The system is expected to hold information of 3,000 or more accounts (more than average CSU to accommodate for growth). This information includes, names, usernames, passwords, payment information, student cards, order

history. During opening hours (7 AM - 7 PM) the system should be able to handle 30 simultaneous users, after this time the system should accommodate for 4 users simultaneously (for testing purposes).

- Start-up time: The mobile system should start-up at around 7 AM to allow for orders to start processing. The website should be available 24/7.
- Shutdown time: The mobile system should shut down at around 7 PM as the cafe will close. The website should not shut down.

- **Supportability**

- Adaptability: The system should be easily adaptable to the platform it supports. e.g. browsers and mobile versions.
- Compatibility: It's expected that the system is compatible with modern versions of Firefox, Google Chrome, Safari, and for mobile at least Android - with the possibility in mind of iOS.
- Level of Support: Level of support will be handled through the user manual which will be constructed to assist those who are having difficulty in using the system. This will include step by step walkthrough and explanation as to why certain field need to be filled e.g. if a user doesn't enter a name and doesn't know why they can't continue even with a prompt, they can go to the user manual which will state where to enter it and why it's needed.
- Scalability: It is expected the system is able to handle 3,000+ accounts efficiently without having issues with space requirements.
- Configurability: It is expected that the system can be modified to adapt to the cafe's needs; this includes configuration of menus and menu items as well as number of seats and logo changes. Configurability is also important because of changing needs from the cafe.
- Maintainability: It is expected that the system supports an admin tool allowing those with administrative access to modify components of the website/mobile app e.g. scheduling updates, updating menu items, discounts and events.

## **System Constraints**

- Design constraints: The system must be able to support both a website and a mobile app meaning it must support all aspects regarding those platforms such as mouse and keyboard support and touch screen support, therefore the system should be created to support touch screen interface.

- Implementation languages: To be determined, Waiting on proposed architecture. Android studio will be used as a software in development of the mobile application. HTML will be used primarily for the development of the website. mysql and sqlite will be utilised for the creation of databases.
- Platform support: This system should be supported on any computer with access to the internet and a generic browser e.g. firefox, google chrome or safari to access the website. The system should also be available on at least modern versions of android to allow access to the mobile app.
- Resource limits: The project only expands 2 subjects (1 year) so it's expected to be finished by the end of session 2 which is in mid-november. It isn't expected that any member of the group puts more than 10 - 12 hours in per week. Monetary requirements should be kept to a minimum if any at all.
- Physical constraints: Screen size constraints for both website and mobile app will have to be taken into account while developing the system, e.g. the app should be able to adjust to what the screen resolution of the phone is.

## **System Interfaces**

- User Interfaces:
  - Look & Feel:
    - Website & mobile: smooth, slick and simple. Quick interactions are important for the efficiency and ease of use of the system. The system should allow for smooth and slick looks to look modern. Maintaining a simple design e.g. minimal buttons and minimal button click will greatly improve the feel of the inference.
  - Layout and Navigation Requirements:
    - Website: The website should be a simple layout and easily navigable using mouse and keyboard inputs, there should be consistency in where buttons are placed per page and simple enough to use that anybody can use it.
    - Mobile App: The mobile app should be simple in design and easily navigable, it must conform to usual seen mobile layouts in which buttons are located at the bottom of each page that direct the flow for the user.
  - Consistency:

- Website: It's expected that navigation controls are always going to be consistent throughout all pages of the website e.g. back button at the top left of page when in a subpage like a menu item. Screen area, size and shape should also be consistent so that the customer is familiar no matter if it's a page they've been to or not. Data presentation such as each item on the menu should all be presented in the same way e.g. 2 menu items are presented with a picture and information regarding the ingredients. Terminology should also stay the same throughout the website e.g. chips should always be referred to as such not on some pages referred to as fries.
- Mobile App: It's expected that navigation controls are always going to be consistent throughout all pages the customer will look at on the app, it will follow common mobile trends e.g. buttons on the bottom of the screen. Screen size and shape should remain the same throughout the app to maintain consistency. data field that the user enters information into should remain consistent to allow familiarity in use, this also goes for presenting data to the user e.g. in the menu each item should be presented the same as the last. Like the website terminology should be kept consistent so as to not confuse the user.
- Interface to External Systems or Devices:
  - Software Interfaces: The system is expected to interact with payment methods so therefore it will have to interact with external services such as EFTPOS, banks, paypal and student cards
  - Hardware Interfaces: Hardware interfaces the system will have to interact with are personal computers and also mobile phones such as android phones. Other hardware interfaces are databases in which the system will interact with to be able to store user information.
  - Communication Interfaces: Communication interfaces that will have to be traversed to deliver this system are internet interfaces and mobile application interfaces.

## **Business Rules**

- Table Booking Requirement Rule 1:
  - Rule 1.1: If the user books table and table is available then allow booking of table.
  - Rule 1.2: If the user books table and table is not available then disallow booking of table.

- Account Creation Requirement Rule 2:
  - Rule 2.1: If the user creates an account and email is correct and not already used and password meets requirements, then the user account is created.
  - Rule 2.2: If the user creates an account and email is not correct or password does not meet requirements then account is not created.
  - Rule 2.3: If the user creates an account and email is already in use then the account is not created.
- Account Login Requirement Rule 3:
  - Rule 3.1: If the user selects login and provides email or name that are valid and password is valid then user logged in.
  - Rule 3.2: If the user selects login and provides email or name that are invalid and password that is valid then the user did not log in.
  - Rule 3.3: If the user selects login and provides email or name that are valid and password that is invalid then the user did not log in.
- Event Booking Requirement Rule 4:
  - Rule 4.1: If the user books an event and that timeframe for the event is available then allow booking of the event.
  - Rule 4.2: If the user books an event and that timeframe for the event is not available then disallow booking of the event.
- Place an Order Rule 5:
  - Rule 5.1: If a user has placed an item into cart and proceeded to checkout and entered valid payment information then order is confirmed.
  - Rule 5.2: If a user has placed an item into cart and proceeded to checkout and entered invalid payment information then order is denied.
- Add to Cart Rule 6:
  - Rule 6.1: If a user has selected the add to cart button then add selected item to cart.