

Beverage Bookers	
Architecture Notebook	Date: <12/04/20>

# Beverage Bookers Architecture Notebook

## 1. Purpose

This document describes the philosophy, decisions, constraints, justifications, significant elements, and any other overarching aspects of the Beverage Bookers system that shape the design and implementation. The main concerns of the project are usability, availability and security. This is reflected in the architecture being chosen, the Three-Layered Architecture. Separate layers of the project allow for a UI to be installed on Android phone to allow users to request orders from the system. The logic layer will check for correct and valid requests and allow data from the database layer to be used or updated to complete the request. The UI layer will then display a result for the user. This functionality allows data to be secure, an increased usability and availability through the UI and reliability in the data sent through the logic layer.

## 2. Architectural goals and philosophy

Beverage Bookers goals and philosophies stem from our high priority non-functional requirements (NFRs) which are:

- **Usability:** It is a major requirement that the Beverage Bookers application is easy to use, easy to understand and allows users to place an order effectively. Actions the user can take should be obvious, easy to understand and quick to use. This means that the user interface (UI) should have clearly labelled buttons, a simple font of appropriate size and easy to change actions made like being able to edit an order after selecting an item.
- **Availability:** The system must be available for it to be used effectively, meaning it needs to be available during the cafe's open hours while also being available for any required maintenance while the cafe is closed. There should be no delays or poor response times that make users have a less than optimal experience.
- **Security:** The system has to be able to keep user information and safe and protected from all other users where only a user can know their own information. It is important this falls in lines of relevant privacy policies and laws. Certain users should be allowed access to different options such as a manager having the ability to customise the menu but prevent unauthorised users from being able to do the same thing.
- **Reliability:** The reliability of the system directly impacts a customers opinion of the cafe and if the application is not reliable, it will not keep or attract any new customers. This could also deter frequent customers or face-to-face customers from trusting the cafe as a whole if they have a particularly bad experience on the application.
- **Audit:** The system needs to keep track of orders due to it dealing with transactions and to keep a record of purchases from customers. The system will also keep track of the status of the order to keep the user up to date on how long they will need to wait for their order.
- **Integrity:** The system requires valid data so that it will be able to rely on accurate and correct databases that keep sensitive data such as user data and payment information safe. If inaccurate and unreliable data is not maintained then issues such as duplicate accounts can lead to users from having inaccurate orders or serious transaction issues occurring.
- **Compatibility:** The application should be able to run smoothly on the user device of choice so it needs to be used only on compatible devices to prevent failed transactions, failed order placements and images not loading which could lead to a negative user experience. The compatible devices should be able to use more modern versions of Android as that is the targeted platform but should also allow for a large potential audience.

Beverage Bookers	
Architecture Notebook	Date: <12/04/20>

### 3. Assumptions and dependencies

#### Dependencies:

- The flexibility of the project team when programming a solution. How well the team can coordinate and effectively solve issues during programming.
- The responses to the questionnaires for barista, cafe manager, customer, kitchen staff and register operator (and if we get answers).
- The ability of the team to roleplay for roles that cannot be filled with an expert in the role like cafe manager.
- The software to create the application is available and hardware (or a simulation of hardware) is available to test it.

#### Assumptions:

- That users understand English as the application will be displayed in English.
- The user has a connection to the server through the Internet.
- The team can code in Java.
- The team has Android devices to test the application out.

### 4. Architecturally significant requirements

- The application has to be able to be edited by a manager to change menu items or state that they are out of stock.
- The system needs to store user information and transaction information while making sure it is all correct and valid data.
- Sensitive data must be stored in a database that is not local to users to keep it safe and protected from users with ill intent.
- The usability of the software will need to be tested by people with minimal / without assistance to check if the application satisfies usability goals.

### 5. Decisions, constraints, and justifications

- A decision made for the architecture is that it has to support Java as a programming language as everyone in the group knows Java and it will boost the programming efficiency a lot if we know the language. The language is also an extremely popular language that allows for issues to be solved from simple google searches and quick troubleshooting.
- A constraint to use Android devices. This constraint was put in place based on a limitation to what devices we own, them being Android devices. This limitation is more of a forced constraint to use Android studio to create our application instead of programming for multiple devices with different software.
- The system is constrained to using a tiered architecture to allow for protecting sensitive data behind an extra layer so that the sensitive data isn't accessible directly by unauthorised personnel.
- The UI must be easily understood, clear and simple to interpret. This is a major requirement for usability but is also important to be accessible to larger audiences. It is critical to have good usability as it creates good experiences for users and can attract new audiences if it is simple and easy to understand.
- A decision to increase usability by increasing font sizes and button size to allow for users to understand the UI easier and help users who have trouble seeing understand the options easier.

### 6. Architectural Mechanisms

#### Ordering Mechanism

The ordering mechanism is for making an order to the cafe and the transaction process when ordering.

Beverage Bookers	
Architecture Notebook	Date: <12/04/20>

Some attributes will be:

- Cart, table
- payment information, (sent to payment system)
- the menu items, objects (listed in the cart)
- price, floating point

The mechanism would start by the user selecting items that are confirmed by the user and added to the cart. They can then check their cart for a total price and option to checkout (purchase). This information is then sent to the payment system for a secure transaction.

### **Admin Mechanism**

The admin mechanism is used by managers / admins who have authorisation to change menus, prices and other details about the menu.

Some attributes will be:

- menu items, objects
- prices, floating point
- menus, lists
- login details (authorised username and password), Strings

The admin can log in and have unique access to modifying the menus by adding and deleting submenus and the items they contain. The prices can also be changed and these changes are reflected onto the menus all users can order from.

### **Table Booking Mechanism**

The table booking mechanism is for booking available tables in the cafe. Some attributes:

- time, Time
- date, String
- number of guests, int
- name, String
- phone number, int
- confirmation, boolean

A customer would start by filling in the booking details and send a request for a manager to schedule the event and respond to the request with the result.

### **Event Booking Mechanism**

The event booking mechanism is used when booking an event at the cafe. Like the table booking mechanism but for booking the cafe for an event, this would be used for clients who are planning a party or who need to fill a lot of orders. Some attributes are:

- time, Time
- date, String
- number of guests, int
- name, String
- phone number, int
- confirmation, boolean

The customer would fill in booking details and have the request looked over by the manager and receive a response of the result.

### **Account Mechanism**

The account mechanism is used when that application starts so that the user can log into their account and check and edit their details. Attributes for this mechanism are:

- username, String
- password, String
- order history, list of orders
- saved payment information, record

The user will be able to check their account details once they sign in. This is access granted only to them

Beverage Bookers	
Architecture Notebook	Date: <12/04/20>

and other users have the same access to their own details but not other users. An account's details can be edited and saved again.

## 7. Key abstractions

- Layer: A separated bunch of code that communicates with other code bunches (layers) to complete a request.
- Request: An action a user makes that is sent through layers of code for processing
- Cart: A list of items that the user selects to purchase from a menu for items.
- Order: The request sent from a user to purchase menu items which they added to their cart.

## 8. Layers or architectural framework

The Three-Layer architecture is an architecture pattern where there are 3 layers of code being a top layer, middle layer and bottom layer.

Top layer: This layer is the User Interface layer, it is where users can input data and is also the layer where they can receive results or responses for their actions. A lot of usability features occur on this layer.

Middle layer: This layer is the Business or Logic layer, a layer for where all business rules and logic occurs. This will check for valid and correct data inputs to prevent incorrect or invalid data from users which can decrease the reliability, integrity and security of the system.

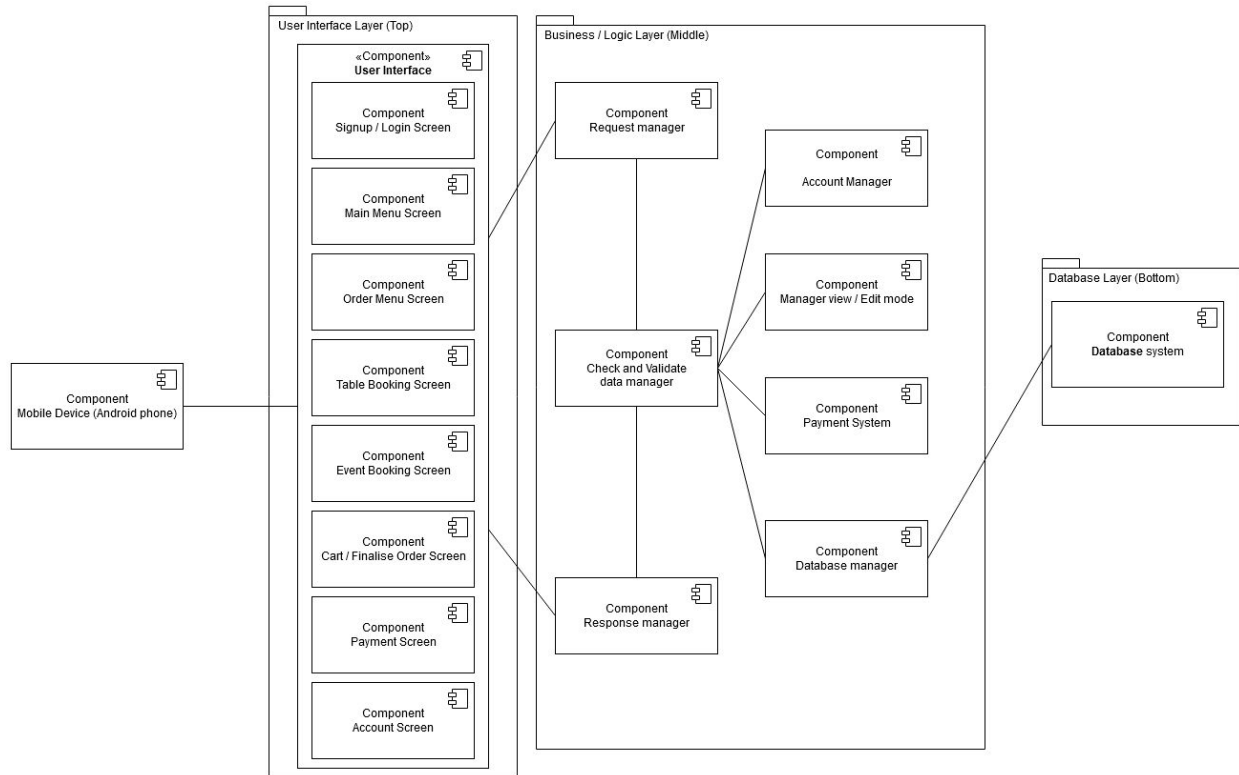
Bottom layer: This layer is the Data layer, this layer contains access to a database for storing sensitive data like transaction data and user data. This is an important layer for integrity and reliability since if the middle layer works correctly it provides correct and valid data that allows users to trust the application and therefore the whole cafe.

Beverage Bookers	
Architecture Notebook	Date: <12/04/20>

## 9. Architectural views

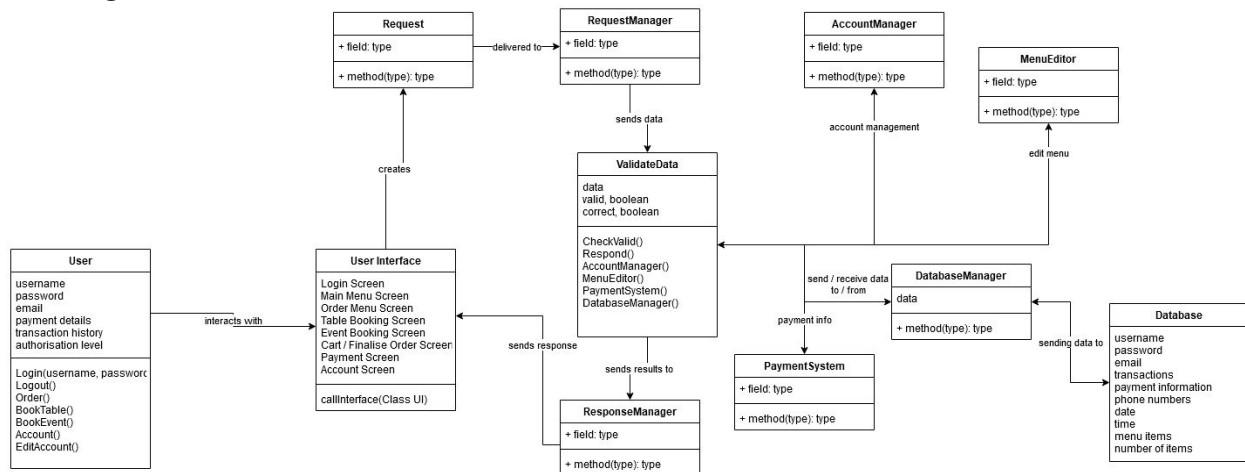
- Logical Views:

### Component Diagram:



Beverage Bookers	
Architecture Notebook	Date: <12/04/20>

## Class diagram:



## Operational Views:

**Creating an account:** 1. The user opens the application through the application button. 2. On the sign in screen that appears, enters the details for their account. 3. The user hits the register button. 4. The register request is sent to the system. 5. The request is checked for valid and non duplicate data. 6a. The request is accepted and the user account is registered to the database. The user can now use the details they provided to log in. 6b. The request is declined and the data is not stored in the database. 7. A response is sent back to the user to tell them the request was accepted or declined based on the result in 6.

**Signing into an account** 1. The user opens the application through the application button. 2. On the sign i screen that appears, the user hits the “already registered? Log in here” button. 3a. The user enters valid account details. 3b. The user enters invalid account details. 4. A user request is sent to the system and checked for valid data. 5a. The data is valid and the system responds by logging the user into the account. 5b. The data is invalid and the system responses to the user with an error message alerting the user the data was incorrect. 6a. The user is sent to the menu screen for options they can perform while logged in.

**Making an order:** 1. Assuming the user has logged in and is on the main menu (follow last case), the user can click on the menu buttons and look through menu items they think are desirable. 2. When an item the user wants to purchase is found, they can click on the item to check details on the item. 3. The menu item can be added to the cart by hitting the “add to cart” button. 4. Repeat 2 to 3 until all items required are added to the cart. 5. The user can check the cart by hitting the “cart” button. 6. The user can check the cart and edit it by clicking the items where they can edit the item (assuming it can be customised) or remove it from the cart with appropriate buttons for the functions. 7. Once the cart is ready to be purchased the user can hit the “order” button to finalise the order and be sent to the payment system to finish the transaction.

**Editing account details:** 1. Assuming the user has logged into an account and is on the main menu (follow signing into an account), the user can use the “account” button to be taken to a new page with details on the user. 2. The user can view their details here but can also hit the “edit” button to change certain fields such as “email”, “password” or “payment details” (if any are saved). 3. Once the user has made any changes required, the “save changes” button will send an update request through the system. 4a. The data is checked and is valid, the data is updated in the database and a “saved changes” response is sent to the user. 4b. The data is checked and is invalid, a response is sent to the user informing them of the data being incorrect or not valid for the field, the data is not updated.

**Booking a table:** 1. Assuming the user has logged into an account and is on the main menu (follow signing into an account), the user can click the “book table” button to start the process. 2. The user

Beverage Bookers	
Architecture Notebook	Date: <12/04/20>

will enter the details like time and number of guests that are coming. **3.** The request is sent to the system to be processed. **4a.** The table booking is valid and is available to be booked with the details the user gave, the system sends a response for an accepted request. **4b.** The table booking either has invalid details or the availability of the table booking isn't available at that time with the current details. The system response with a declined request response. **5a.** An accepted table booking leads into the order system.

**Booking an event:** **1.** Assuming the user has logged into an account and is on the main menu (follow signing into an account), the user can tap the "book event" button to start the process. **2.** The user will enter the details of the event like time and number of guests that are coming. **3.** The request is sent to the system to be processed. **4a.** The event booking is valid and is currently available at the cafe with the current details that the user gave, the system sends an accepted request response to the user. **4b.** The event booking either has invalid details entered or currently isn't available at the cafe with the current details provided, the system responds to the user with a declined request response. **5a.** An accepted event booking leads to the order system.

Beverage Bookers	
Architecture Notebook	Date: <12/04/20>

• Use case: [Link](#)

Beverage Booker App

