# MAD Assignment Design Documentation

# 1 Introduction

## 1.1 Purpose of App

The main purpose of this documentation is to show the background, functionalities, design structure in this App of Cup Cake Bakery (CCB)

## 1.2 Background of App

As we know, cupcakes are the very important along with the main meals. The demands of the cupcakes are dramatically increasing these years. Based on the recent research, it is found that the customers want a more advanced convenient way to buy the cupcakes instead of going to the cupcake bakery in person. And this really drives the appearance of this App. CCB is to provide the customers an strong and widely used online market platform and let the customers taste any kind of cupcakes as soon as possible.

# 2 Functionality Requirements

## 2.1 System Functionalities (SF)

| SF_ID | SF_NAME | SF_DESCRIPTION | SF_PRIORITY (0-10) |
|---|---|---|---|
| 2.1.1 | Show overall cake list | This is where the customer starts the trip of buying cupcake online. The function will download the current cupcakes online and show them for the customer to view and pick up the cake they like | 10 |
| 2.1.2 | Show selected cake details | This function will further show the more details about the selected cake, including cake size, | 10 |

| | | | |
|---|---|---|---|
| | | cake view, cake name and so on | |
| 2.1.3 | Enable cake buying | This function starts the process of purchasing the selected cake. This process includes delivery selection and payment completion | 10 |
| 2.1.4 | Enable Delivery Selection | This function will show two choices of delivery ways, one is taking away in person in the outlet and the other one is transport delivery. For transport delivery, the customer will be asked to input the personal details to help complete the delivery and this includes address, name, post code and so on. | 9 |
| 2.1.5 | Enable Order Payment | This function gives the customer the way to pay the selected cake. The customer will have to provide the account info to help complete the payment | 10 |
| 2.1.6 | Show paid orders records | All the paid order by the customer will be stored in the local database for future review. The shown details include the bought cake details, purchase date and so on.<br>If there are no the paid orders currently, the empty message | 10 |

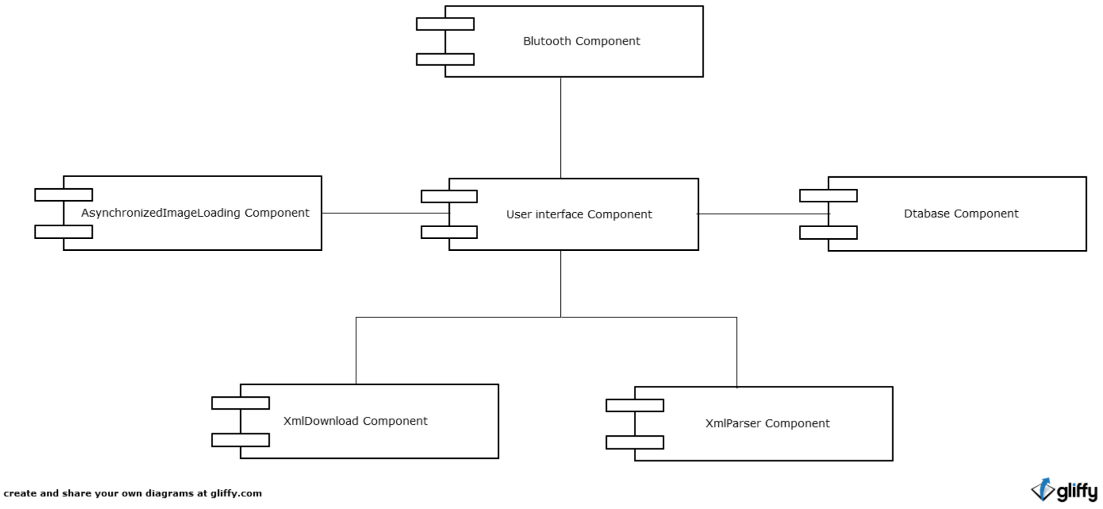
| | | | |
|---|---|---|---|
| | | will be shown | |
| 2.1.7 | Delete unwanted order records | The customer can have chance to delete any order record that they do not want to keep any more. | 8 |
| 2.1.8 | Store the paid order | When the customer finishes payment, the paid order shall be stored in the local database. | 8 |
| 2.1.9 | Enable favorite cake selection | The customer can setup their favorite cake from paid orders | 8 |
| 2.1.10 | Store favorite cake | When the particular cake is setup as favorite cake, it shall be stored in local database for future use. | 9 |
| 2.1.11 | Show favorite cakes | All the favorite cakes details shall be shown for the customer<br>If there are no the favorite cakes currently, the empty message will be shown | 10 |
| 2.1.12 (not completed because of time limit) | Delete favorite cake | The selected favorite cake shall be deleted from database and the current view on the phone. | 8 |
| 2.1.13 | Enable favorite cake share via Bluetooth | The system shall be able to provide Bluetooth connection and data transfer | 10 |
| 2.1.13 | Enable Bluetooth server | The receiving customer can play a role of Bluetooth server to receive the data from client | 10 |
| 2.1.13 | Enable Bluetooth client | The sending customer can play a role of | 10 |

|  |  |  |  |
|---|---|---|---|
|  |  | Bluetooth client to connect and send the data to the server |  |
| 2.1.14 | Update favorite cakes list view | The current favorite cakes view on the screen shall be updated once the sharing finishes. | 10 |
| 2.1.15 | DIY cakes | The customer can DIY the sleeted cake via changing the cakes size, cake accessories, cakes seasonings and so on. | 10 |
| 2.1.16 | Update cake details after DIY | The cake that is DIY by the user shall be updated and these updated details include cake price and DIY items. | 10 |

# 3 Design

## 3.1 Component Design

### 3.1.1 Overall Component Diagram (OCD)



### 3.1.2 Explanations of Overall OCD

**XmlParser Component**

-Belong to model Layer

-Involves a class of XmlParser.java

-Declare objects of list of cakes that is used to hold the different cakes created based on the cake

tag in cakes.xml file
-Declare object of diy that is used to hold the different materials that can be changed by the user
-Will be used by the service along with user interface component to start parse the downloaded xml from server and manipulate the result in warehouse object for future use

**XmlDownload Component**
-Belong to model layer
-Involves a class of XmlDownloader.java
-Download the cakes.xml from server
-Translate the cakes.xml to string
-Will be used by the service along with user interface component to start downloading xml from server

**User interface Component**
-Belong to Controller and View Layer
-Involves 11 different activities:
android:name="com.cupcake.view.CakeAssortment"
android:name="com.cupcake.view.SelectedCake"
android:name="com.cupcake.view.Payment"
android:name="com.cupcake.view.DeliveryWay"
android:name="com.cupcake.view.EndPayment"
android:name="com.cupcake.view.SuccessPayMessage"
android:name="com.cupcake.view.Accessories"
android:name="com.cupcake.view.DiyCake"
android:name="com.cupcake.view.CakeRecords"
android:name="com.cupcake.view.FavouriteCakes"
android:name="com.cupcake.view.BluetoothRemoteDevicesList"

**AsynchronizedImageLoading Component**
-Belong to model Layer
-Involves class of AsyImgLoad.java and interface of callback.java
-This component will be running at another thread and cooperate with user interface component closely
-Asynchronized download the cake images based on the image path of image path tag in cakes.xml
-Will be used by user interface component to update the gallery cakes list

**Bluetooth Component**
**-** Belong to model Layer
-Involves one class of BlueToothManager.java
-This component includes 3 different threads to relatively deal with active connection as client, passive accept as server and data transfer between client and server connected sockets
-Will be used by user interface component to update the favorite cakes list

**Database Component**
-Belong to model Layer
-Involves one class of Database.java
-This component will handle the order table updating, deleting and inserting the paid orders
-Will be used by user interface component to handle the persistence of cake object and order in local mobile phone device

# 4 Testing

4.1
The testing tools used in this project are java unit testing frameworks and java logging class functions.

4.2
The main testing kinds in this project are unit testing, integration testing and system testing.

Unit testing is used to test some of important methods. For instance, the database's insert, update and query functions are tested using unit testing.

Integration testing is applied when the particular component is completed. And it will used to test how well the new component interacts with previous ones. For instance, the Bluetooth component was tested with the calling activity of FavouriteCakes.java and it is tested that the data from the Bluetooth component are normal and the activity can receives the correct data.

System testing happens when all the components re done and they are put together to test. This will show whether the overall system works fine. This is the final testing in this project.

# 5 Conclusion

In conclusion, the CCB project provides the customer a faster and more featured way to buy and DIY the cupcake online. Particularly, the DIY function gives the customer more chances to make the cake that really want. Another feature of CCB is the share of cakes with friends via Bluetooth. This can let the customer interact with other customers and share the happiness with others.

Perhaps the most difficult part in this project is the Bluetooth implementation. This is because the drivers API for different brands of mobile phones are incompatible and some of them are hardly work in some of phones. In addition, the Bluetooth connection is easily effected by outside factors such as other radios in the air, long distance and weak signal strength. Furthermore, the security issue within Bluetooth is also concerned. When the device acts as server and become discoverable by other devices, it is easy to leak the security details by the illegal connection with the device.