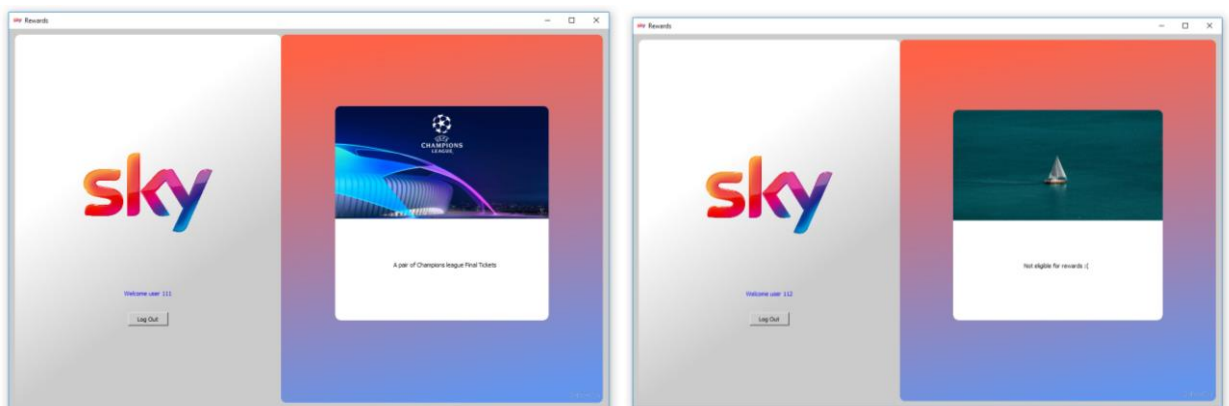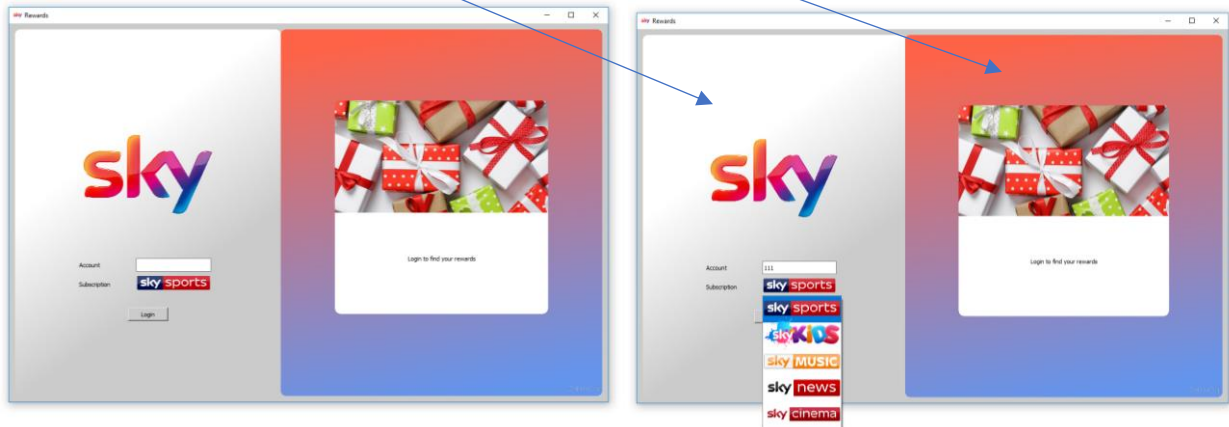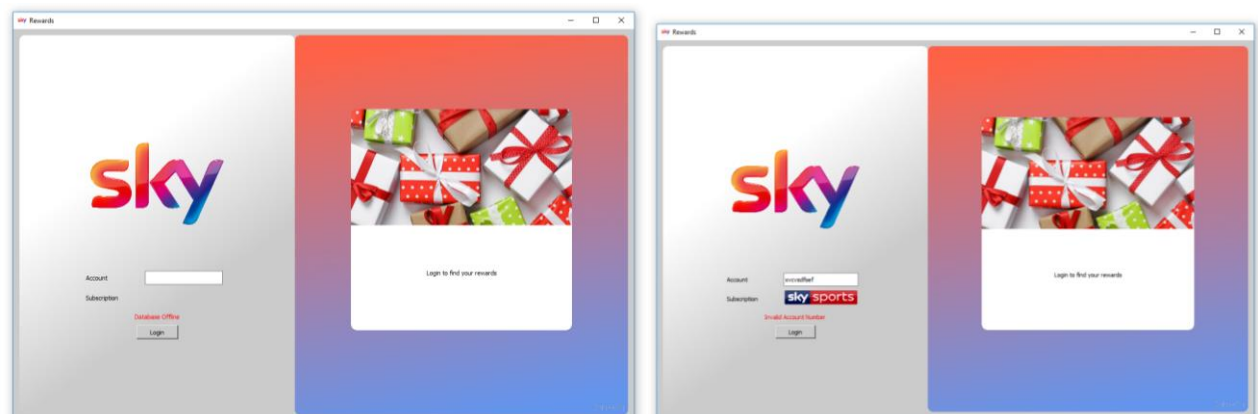# Sky Reward App

User Interface

2 Widgets: one on the left for user input and one of the right for output.

- When correct Account number and correct subscription is entered, user is shown welcome message and reward

- When correct Account number and incorrect subscription is entered, user is shown welcome message but no reward

- When the SQLite database file (.db file) is missing, error message is shown. **Please copy "accounts.db" to the current folder of the application for the app to run correctly.**

- When incorrect Account number and correct subscription is entered

Classes

1. Custom_graphicsview : shows the reward image and text in a rectagle with rounded corners
2. Custom_combobox : shows the list of available subscriptions as large images
3. Database : manages the SQLite database for user accounts and subscription details
4. Login : custom widget to take input from the user
5. Reward : custom widget to show reward to the user
6. Mainwindow : main window of the app

Data

Data for user accounts ard subscription details are stored in SQLite .db file with two tables, one for user accounts and another for subscription and reward details. **Please copy "accounts.db" in the current folder of the application for the app to run.**

Table: Accounts

	ID	AccountNumbers	ChannelID
	Filter	Filter	Filter
1	1	111	1
2	2	112	2
3	3	113	3
4	4	114	4
5	5	115	5

- Account table for user account number and subscription / channel

Table: ChannelsRewards

	ChannelID	Name	Icon	Reward	RewardPictures	BackgroundPictures
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Sky Sports	BLOB	A pair of Cha...	BLOB	BLOB
2	2	Sky Kids	BLOB	Not eligible fo...	NULL	BLOB
3	3	Sky Music	BLOB	Karaoke Pro ...	BLOB	BLOB
4	4	Sky News	BLOB	Not eligible fo...	NULL	BLOB
5	5	Sky Cinema	BLOB	Pirates of the ...	BLOB	BLOB

- Channel table for subscription and reward details.

Unit testing

The app was tested using Qt Test framework. Created a test for each classes. Some of the classes could not as Custom_graphicsview and Custom_combobox were not tested directly as only "paintEvent" method was reimplement from their base class "QWidget".

1. `void` **`test_mainWindow`**`();`
2. `void` **`test_database`**`();`
3. `void` **`test_globalVariables`**`();`
4. `void` **`test_login`**`();`
5. `void` **`test_reward`**`();`

Since all of the logic are applied in "`void` **`test_login`**`();`" and "`void` **`test_reward`**`();`" this is testing by entering known user account number as inputs and check if the outputs are produced correctly.