**UI​​Development**: The project is created with JavaFX. There are three .fml documents which display the Entry, Budgeter, and Info Views.

**Architecture:** Model {​​ AccountGroup.java, Account, Countable ​​}

Views {​​ Entry.fxml, Budgeter.fxm }

Controllers{ EntryController.java, BudgeterController​​ }

1. Object oriented elements that you write the code for:
   1. Classes

**AccountGroup.java, Account.java**

* 1. Subclasses

**BugetController.java of Switchable**

**EntryController.java of Switchable**

**Expense.java of Countable**

**Income.java of Countable**

* 1. At least one abstract class

**Countable.java, Switchable.java**

* 1. At least one interface

**Catergorizable.java**

1. Code elements that you utilize:
   1. One or more collection classes

**AccountGroup.java has an ObservableList<Account>**

**Account.java has an ObservableList<Countable>**

* 1. Exception Handling

1. The application must have a clearly defined model (as in the M in MVC).

**The model that is used is the AccountGroup.java. It contains all other objects and is what is displayed on the Budgeter View**

1. The UI must utilize multiple scenes and at least one of the scenes must contain of the scene graph changed based on the application state.

**There are two scenes (Entry,and Budgeter)**

**Budgeter has a changes state on lines**

**Budgeter has different effects based on state on lines**

1. There must be a way to access “About” information that includes information about you and the application

**The entry page has information about myself and the app. You can access this by starting the app or pressing the back button on the Budgeting page.**

1. The application must save data and load data. The target for saving/loading data can be files, a network service, and/or a database.

**The app uses serialization in order to save and load data. There are custom implementation of readObject() and writeObject() in AccountGroup.java, Account.java, and Countable.java, all of which implement Serializable (Implementations are the last two methods in each Class)**