**Joshua Bell**

**3537603**

**SWE 4403**

**Final Project**

**Software Application**

The software system implemented for the term project is a text editor. The user will be able to create, open, and modify text documents.

**Software Architecture**

The system implements the layered architectural pattern. The system is divided into two layers: the UI layer, the System layer. The layers contain the following classes:

**UI Layer**

* MainApp
* OpenDocumentCommand
* HelpWindowSingleton
* UserDirector
* ChatRoom

**System Layer**

- MementoAction

- BoldAction

- UndoAction

- SaveAction

- FindAction

- DataUpdateAction

- PathUpdateAction

- Document

- Decorator

- BoldDecorator

- Façade

- Convert

- SaveAccordingly

- DetermineType

- UserDirectory

- User

- ChatRoomMediator

- Subcriber

- NonSubcriber

- Subject

- ConcreteObserver

- GreetingOnePrototype

- GreetingTwoPrototype

- HelperFunctions

- FindIterator

- ProxyImage

- RealImage

- Originator

- Memento

- CareTaker

- DocumentCreator

- TextDocument

- DocumentFactory

**Software Design Patterns**

1. **FactoryMethod** - The factoryMethod design pattern was used for the createDocumentFactoryMethod(). Right now it is only used to create Text documents but I intend to add HTML documents to.
2. **Command** - The command design pattern is used to create command objects for OpenDocumentCommand. Each command object has a “excute”.
3. **Iterator** – The iterator design patter is used to iterate over the words in the text file. In my application it is used to find a particular word.
4. **Singleton** – This is used to show the help window, only one help window should exists.
5. **Decorator** – This design pattern is used to decorate a document component which can decorate the text so it is bold.
6. **Prototype** – This design pattern is used to greet the user when the greeting button is clicked. It creates a prototype of greeting.
7. **Façade** – This design pattern has been implemented to determine whether a document is a text, html and save accordingly. (SAVE BUTTON)
8. **Proxy** – This pattern is used to reduce the memory footprint of RealImage object loading, mine is implemented with strings indicating when a load is talking place. When you press the Proxy MenuItem it calls display twice but only loads the image once.
9. **Memento** – This design pattern allows you to undo to the last save of the session.
10. **Builder** – This design pattern is used to build a new user.
11. **Mediator** – This design pattern mediates a chatroom between two users which have been hard coded they do not relate to the users built by the builder.
12. **Obserever** – This design pattern is created to tell all the observers of a state change when the observer button is pressed.