**Detailed Description for the High-Level design**

**User Profile**

The system is using SQLAlchemy to access and manage data in the SQLite database. SQLAlchemy is adapted into pythonic domain language.

**Template package**

The template package includes two html files, which are UserProfile.html and Main.html. those files are all written in html and using jinja to add control statements including url\_for(), render\_template():

**-url\_for() will build a url for a specific function.**

**-render\_template() will render a HTML file which base on the jinja and html code.**

In this Templates package, we use ninja’s template inheritance feature, which allows us to move the parts of the page layout that are common to all templates and put them in a main template from which all other templates are derived. Also, those templates will handle message by using MessageHandler in the user profile **js package .**

**Controllers package**

The Controllers package consists of three parts of the system, which are Direct Messaging and User Profile. Those are written in python programming language.

**Direct Messaging** has a post() method and a url\_for() method:

**-The post() method tells the server that it wants to post some new information to that url and that the server must ensure the data is stored and only stored once.**

**-url\_for() method will redirect a user to another webpage.**

**User Profile** has a get() method:

**-The get() method will call the getchat() method in Services package and return the history chat of a user.**

The **routes.py package** file imports the methods for each of the GET and POST requests that the user asks for. Also, in this package, it will be using Flask-SocketIO package to give flask applications access to low latency bi-directional communication between the clients and the server.

**Sevices package**

This package is use to dispatch a message from a user to another user. Also, the getchat() method will return a history chat of a user.

**-dispatch(message) method will dispatch a message from a user to another user**

**-getchat(userId) method will get the history chat below the user information.**

**SQLALchemy**

In the SQLALchemy python code file, Session is established to connect with the database and represents for all objects you’ve loaded or associated with it during the lifespan. the Session includes add() method, commit() method, delete() method.

**-add() is use to place instances in the session.**

**-commit() is use to write changes to the database.**

**-close() will safely dispose of any remaining session transaction objects associated with the session.**

**Bootstrap package**

The Bootstrap package is use for cascading style sheets. It describes how HTML elements are to be displayed on screen, paper or other media.

**Models package**

This package includes a message database file. The database file creates a table for a message, which includes:

**-id: integer**

**-body: String**

**-sender: String**

**-recipient: String**

**-Date: date and time**