Boğaziçi University

Software Engineering

SWE 573 – Software Development Practice

Spring 2017

Term Project

Software Design Document

SENTWORT

Prepared By:

Umur Türkay

umur.turkay@gmail.com

1. Introduction

The Software Design Document is a document to provide documentation which will be used to aid in
software development by providing the details for how the software should be built. Within the Software Design Document there are both written explanations and graphical documentation of the software design for the project including use case scenarios, sequence diagrams, object behavior models, and other supporting requirement information.

1.1. Purpose

The purpose of the Software Design Document is to provide a written and graphical description of the system design which provides appropriate information to build the system.

1.2. Scope

This report defines the high level design of and technological decisions to build the Sentwort web application.
This document defines and describes the use of each view, the functional requirements, use-case realization, the layers and subsystems of the application with appropriate diagrams and explanations. It is focused on main and critical parts of the system.

1.3. Definitions, Acronyms and Abbreviations

Sentwort: The web application which is the subject of this project.

Hashtag: a type of metadata tag used on social network and microblogging services (Twitter for this application) which makes it easier for users to find messages with a specific theme which starts with # character.[1]

Sentiment Analysis: the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral.[2]

1.4. References

[1] https://en.wikipedia.org/wiki/Hashtag
[2] https://www.google.com.tr/search?q=what+is+sentiment+analysis
[3] https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html
[4] http://www.telerik.com/jsp-ui
[5] https://dev.twitter.com/rest/public/search
[6] http://www.sentiment140.com/

2. Use Cases

2.1 Actors

2.1.1 User

User is a generic user of the Sentwort application. User may be anybody who has internet connection and a laptop/PC which has a web browser to use the application.

2.2 Use Case Scenarios

- **Login Use Case**

  Actors User

  Description User logins to the system

  Flow of events

  1. User enters Sentwort web page URL to browser's address bar
  2. User accesses login page
  3. User enters their username and password
  4. User clicks "login" button
  5. User logins to the system and gets redirected to the home page

- **Register Use Case**

  Actors User

  Description User becomes a registered user

  Flow of events

  1. User enters Sentwort web page URL to their browser's address bar
  2. User accesses login page
  3. User clicks "Register" button
  4. User accesses registration page
  5. User enters username, password, confirmation for password, secret question and secret answer for registration
  6. User clicks "Sign Up" button
  7. User is registered and gets redirected to the home page

- **Forgotten Password Use Case**

Actors User

Description User sets a new password

Flow of events

1. User gets error while logging in to the system about wrong username and/or password
2. User cannot remember their password
3. User clicks “Forgot your password?” link
4. User enters username and clicks "Confirm"
5. User sees their secret question, enters the appropriate answer.
6. User clicks "Set new password"
7. User is prompted to set a new password
8. User sets a new password and gets redirected to home page

- **Add Hashtag to be Listened Use Case**

  Actors User

  Description User adds an hashtag to the system to be listened

  Flow of events

  1. User logins to the system
  2. User gets redirected to home page
  3. In the left menu, user clicks "Hashtag Management"
  4. User enters an hashtag to the "Hashtag" textbox
  5. User clicks "Add"
  6. User sees the hashtag added to the list

- **Start/Stop Listening an Hashtag Use Case**

  Actors User

  Description User starts/stops the system to listen an specific hashtag

  Flow of events

  1. User is already logged in to the system and is on "Hashtag Management" page
  2. User selects an hashtag from the table
  3. User clicks "Start Listening"/"Stop Listening"
  4. Hashtag is active/passive

- **See Sentiment Report Use Case**

  Actors User

  Description User displays sentiment report and filters the data

  Flow of events

  1. User is already logged in to the system
  2. User clicks Sentiment Report" from the left menu
  3. Account of the tweet, tweet itself, hashtags, links, sentiment of the tweet and share date appears as a table
  4. Users sees the data of last month at the beginning
  5. User enters filters and clicks "Search"
  6. Table is updated according to the selected filters

- **See Hashtag Report Use Case**

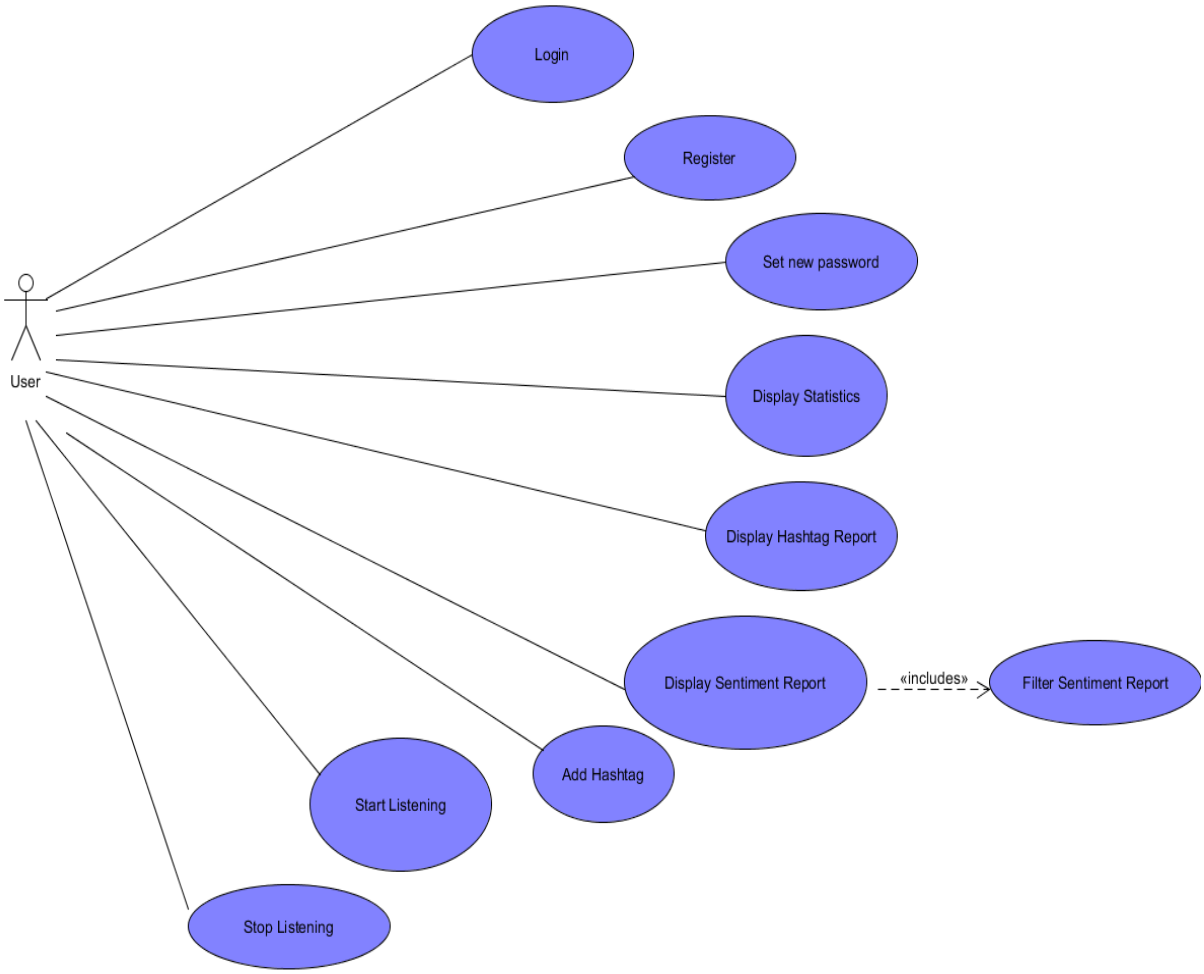
  Actors User

  Description User checks out statistics about hashtags defined in the system

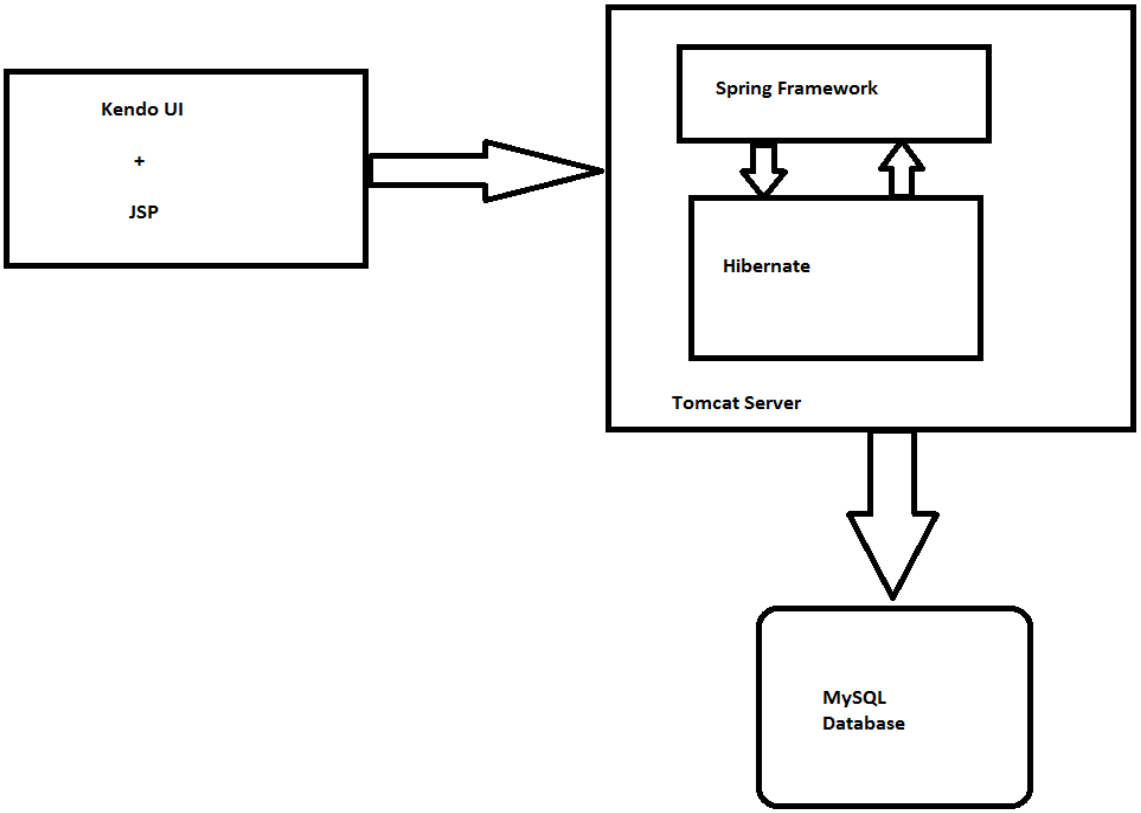
  Flow of events

  1. User is already logged in to the system
  2. User clicks "Hashtag Report" from the left menu
  3. Number of tweets, number of tweets analyzed sentimentally, number of positive, negative and neutral tweets, start date of listening and status of hashtag (active/passive) appears on the page

2.3 Use Case Diagram



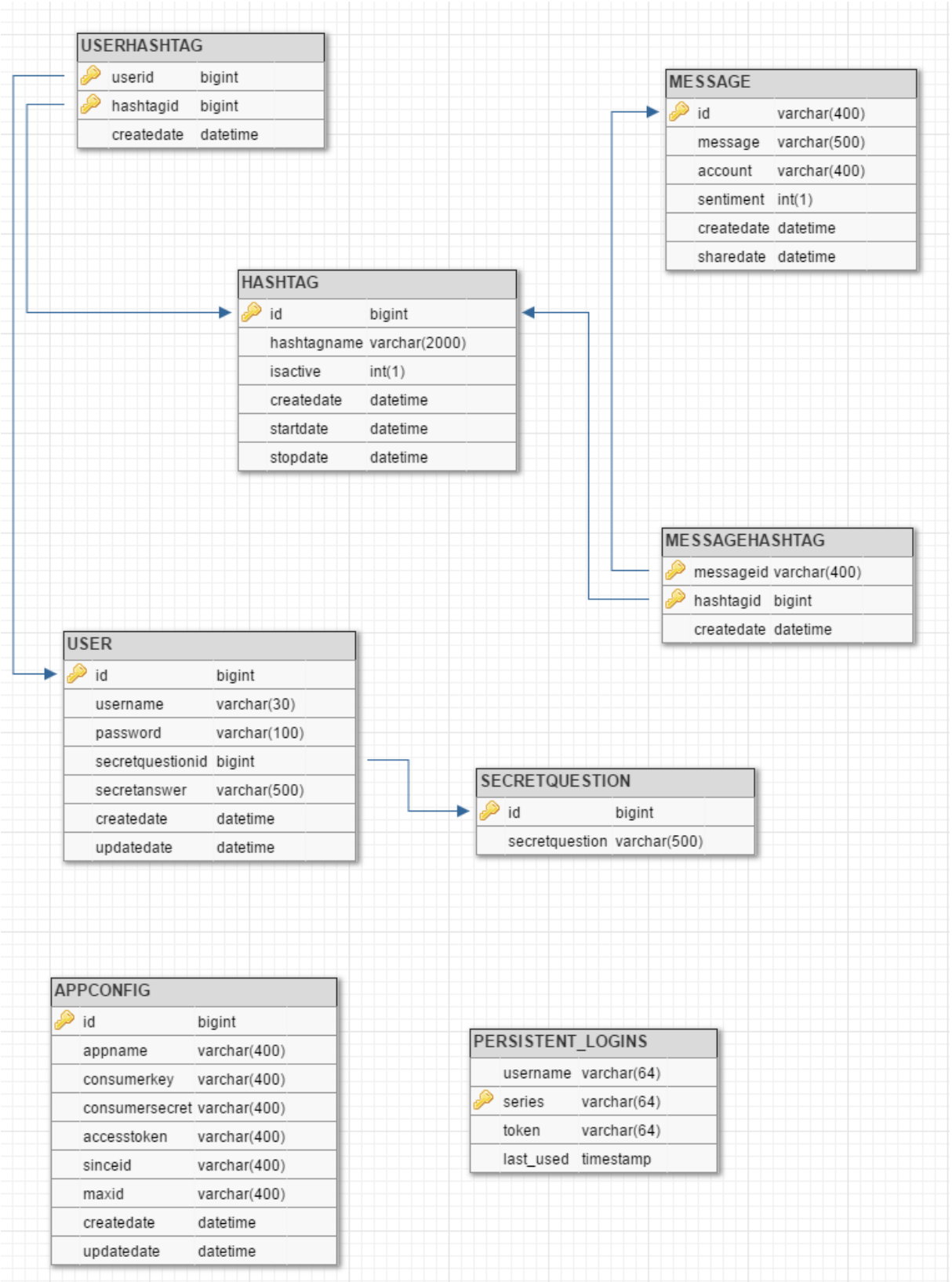
3. System Overview



Spring MVC Framework is used for implementation. It is a web application framework which lets you implement your web application according to the model-view-controller design pattern. [3]

For view part, Kendo UI for JSP [4] is decided to be used. It is a user interface framework for building web applications. As the database, MySQL is the choice.

4. Data Model

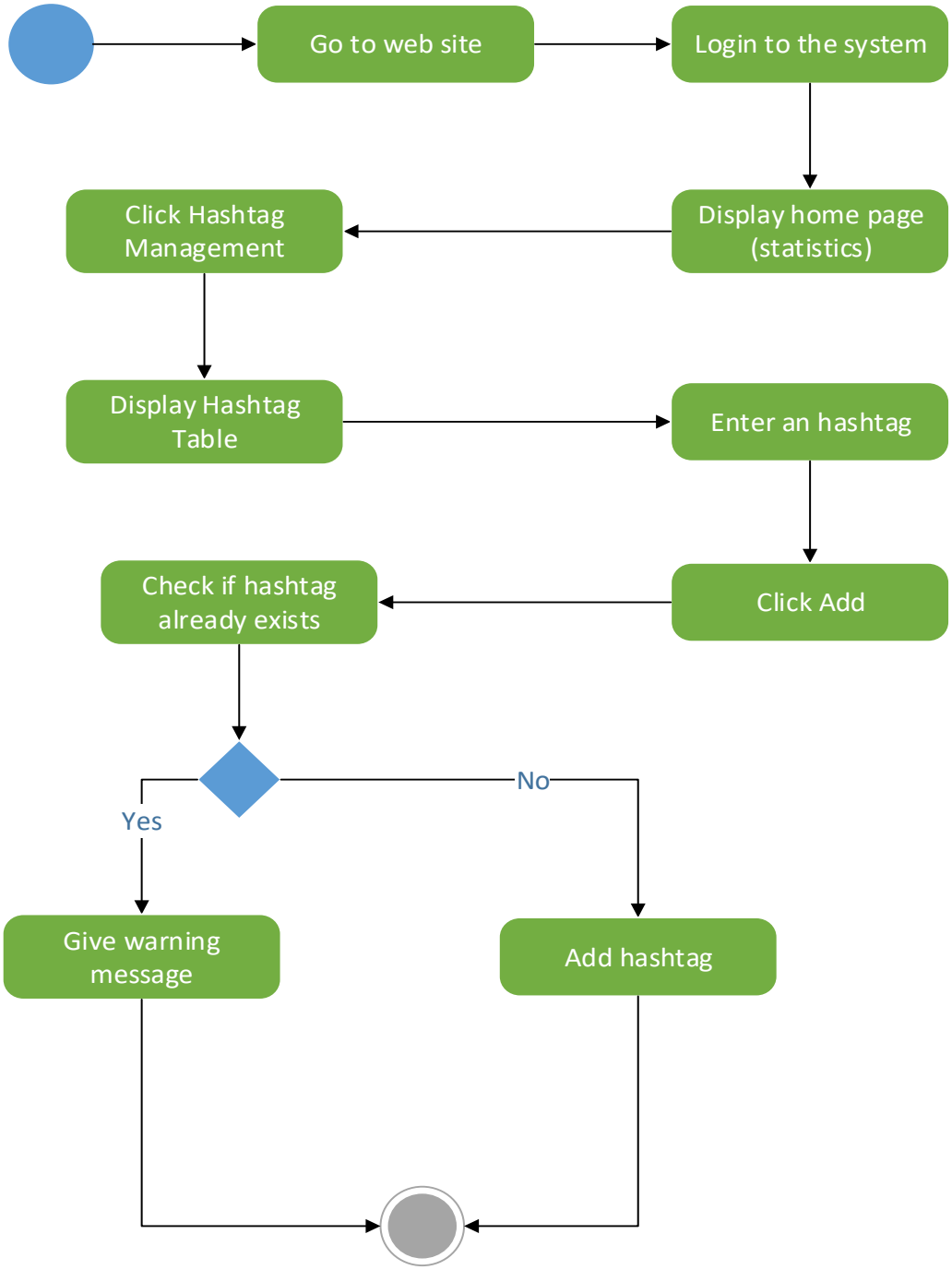

USERHASHTAG
userid bigint
hashtagid bigint
createdate datetime
MESSAGE
id varchar(400)
message varchar(500)
account varchar(400)
sentiment int(1)
createdate datetime
sharedate datetime
HASHTAG
id bigint
hashtagname varchar(2000)
isactive int(1)
createdate datetime
startdate datetime
stopdate datetime
MESSAGEHASHTAG
messageid varchar(400)
hashtagid bigint
createdate datetime
USER
id bigint
username varchar(30)
password varchar(100)
secretquestionid bigint
secretanswer varchar(500)
createdate datetime
updatedate datetime
SECRETQUESTION
id bigint
secretquestion varchar(500)
APPCONFIG
id bigint
appname varchar(400)
consumerkey varchar(400)
consumersecret varchar(400)
accesstoken varchar(400)
sinceid varchar(400)
maxid varchar(400)
createdate datetime
updatedate datetime
PERSISTENT_LOGINS
username varchar(64)
series varchar(64)
token varchar(64)
last_used timestamp

5. Dynamic Model

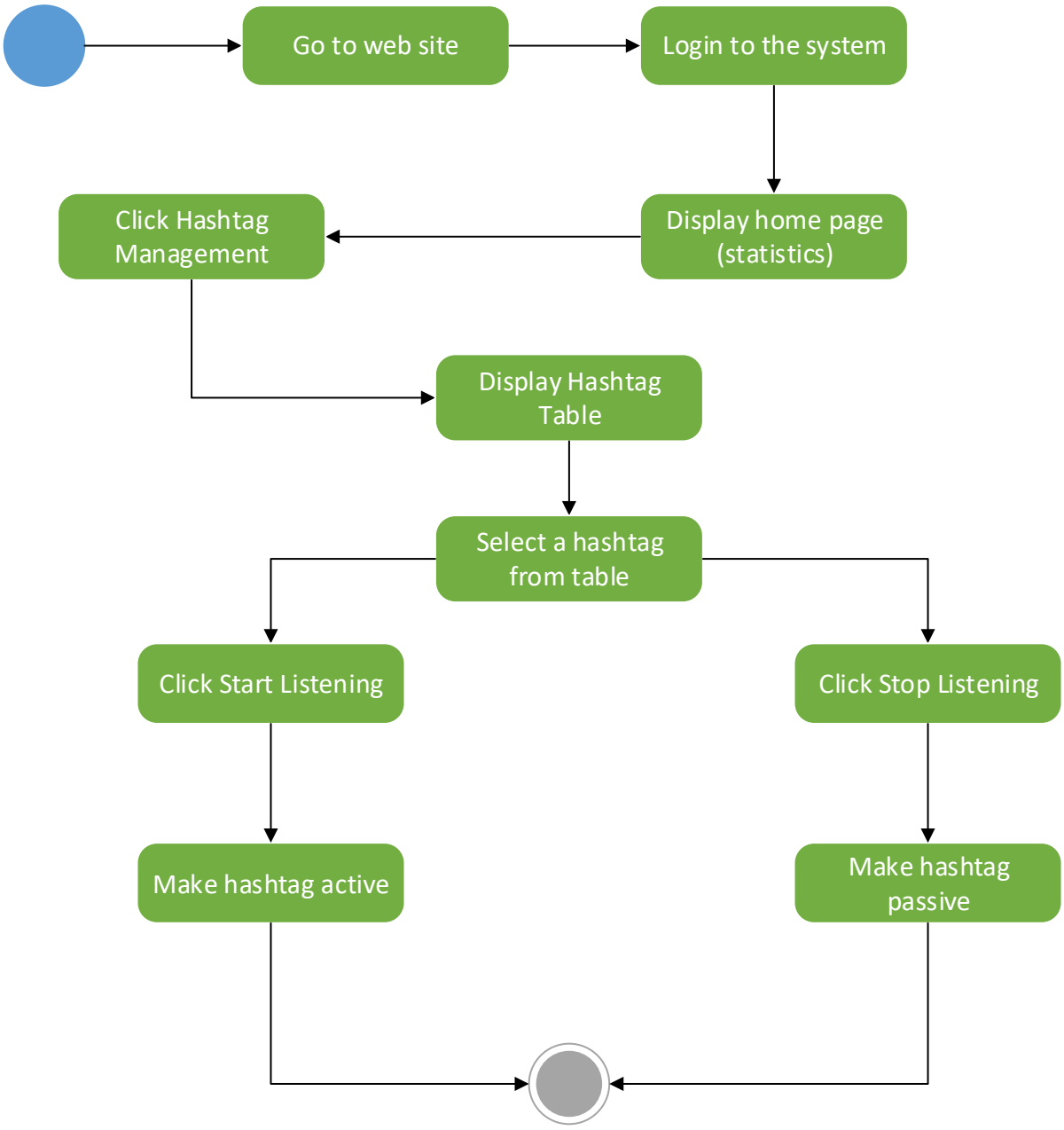
5.1 Hashtag Management

Users of the application can add hashtags to be listened by the system. Application retrieves tweets containing hashtags defined by the user from Twitter Search API[5] by sending request for every minute using quartz jobs. System also analyzes sentiments of the tweets using Sentiment140 Twitter sentiment analysis tool[6].

Activity Diagram for hashtag add:



Activity Diagram to start/stop listening a hashtag:



5.2 Sentiment Report

Users of the application can display sentiment report based on the data retrieved from Twitter. Users can filter out the result by share date of the tweet, hashtag added by themselves and by sentiment.

Activity Diagram: