# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 25/03/14 | 1.01 | Model component added, diagrams added | Umut Uzgur |
| 25/03/14 | 1.02 | Controller component added, diagram added | Cholpon Abdyzhaparova |
| 27/03/14 | 1.03 | Login activity diagram added, model classes deleted | Umut Uzgur |
| 29/03/14 | 1.04 | Controller class diagram replaced | Cholpon Abdyzhaparova |
| 30/03/14 | 1.05 | View Component and related diagrams added | Uğur Özkan  Burak Tutanlar |
| 30/03/14 | 1.06 | Workflow diagrams added for Controller | Cholpon Abdyzhaparova |
| 31/03/14 | 1.07 | Software Architecture and Design Description added with diagrams. Descriptions added/corrected. Document organized. | Çelebi Murat |
| 1/04/14 | 1.08 | Controller is renamed to Manager and divided into two parts | Cholpon Abdyzhaparova |
| 1/04/14 | 1.09 | Component names and diagrams are improved. | Cholpon Abdyzhaparova, Burak Tutanlar, Çelebi Murat |

**TABLE OF CONTENTS**

1 Introduction 3

1.1 References 3

1.1.1 Project References 3

2 Software Architecture Overview 3

3 Software Design Description 4

3.1 DataRequester 5

3.1.1 Component Interfaces 5

3.1.2 Component Design Description 5

3.1.3 Workflows and Algorithms 6

3.1.4 Software Requirements Mapping 9

3.2 GUIManager and RequestManager 10

3.2.1 Component Interfaces 10

3.2.2 Component Design Description 10

3.2.3 Workflows and Algorithms 12

3.2.4 Software Requirements Mapping 13

3.3 GUI Component 13

3.3.1 Component Interfaces 13

3.3.2 Component Design Description 14

3.3.3 Workflows and Algorithms 15

3.3.4 Software requirements mapping 16

4 COTS Identification 16

# Introduction

This document describes the design of the TDS software system.

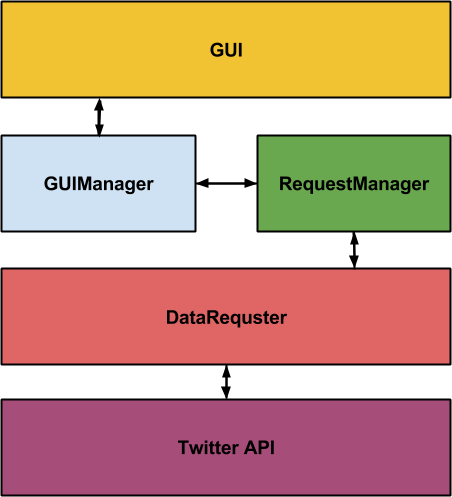
## References

### Project References

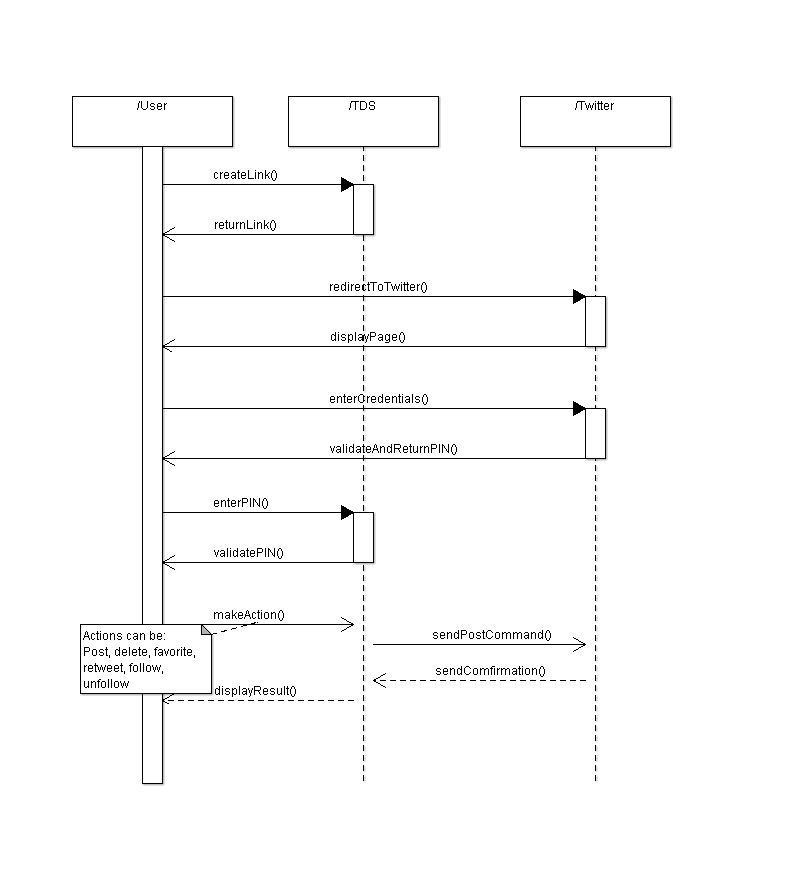
| # | Document Identifier | Document Title |
| --- | --- | --- |
| 1 | Doc 1 | Software Development Plan |
| 2 | Doc 2 | Software Requirements Specifications |

# Software Architecture Overview

We are utilizing classical Model-View-Controller pattern, thus we have three major components. GUI part contains only View component. Other two components, which are Model and Controller, are inside Front-end Services. TDS provides the user with a way to interact with Twitter services, so basically it acts as a messenger.



**Figure 1:** System Architecture



**Figure 2:** Sequence Diagram

# Software Design Description

TDS is a small software and has low complexity. In its core it has three main components.

* **GUI:** It is responsible for displaying a given formatted data. Formatted data can be a string object or an image object. Also it directs user actions to Controller component. (see Figures 12 and 14)
* **DataRequester:** This component is the middle-man between the TDS and Twitter API. It listens for any request from Controller component and tries to answer. It has 4 handlers that does the actual communication (see Figures 3, 4, 6 and 7).
* **GUIManager and RequestManager:** GUI Manager is a component that speaks to GUI and RequestManager. GUI delegates the action listener implementation to GUIManager. After getting signal of action listener from GUI, GUIManager tells the RequestManager what to do. DataRequester Manager gets commands from GUIManager and tells DataRequester what data to give and what action to do. (see Figures 9, 10)

## DataRequester

### Component Interfaces

NavigationHandler

* In charge of navigating through the user profiles
* Responsible for getting the user timeline, mentions, favorites, followers, followings, tweets, to follow and to unfollow

TweetHandler

* In charge of tweeting, retweeting, to favorite a tweet and to delete the user's tweet

DmessageHandler

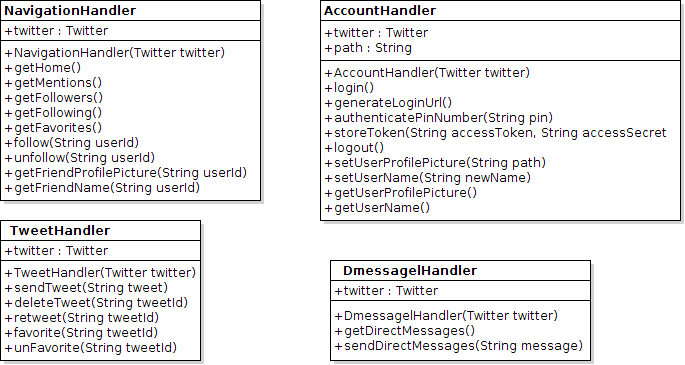
* Responsible for sending direct messages, getting the direct message inbox

AccountHandler

* Responsible for changing the user name and user profile picture

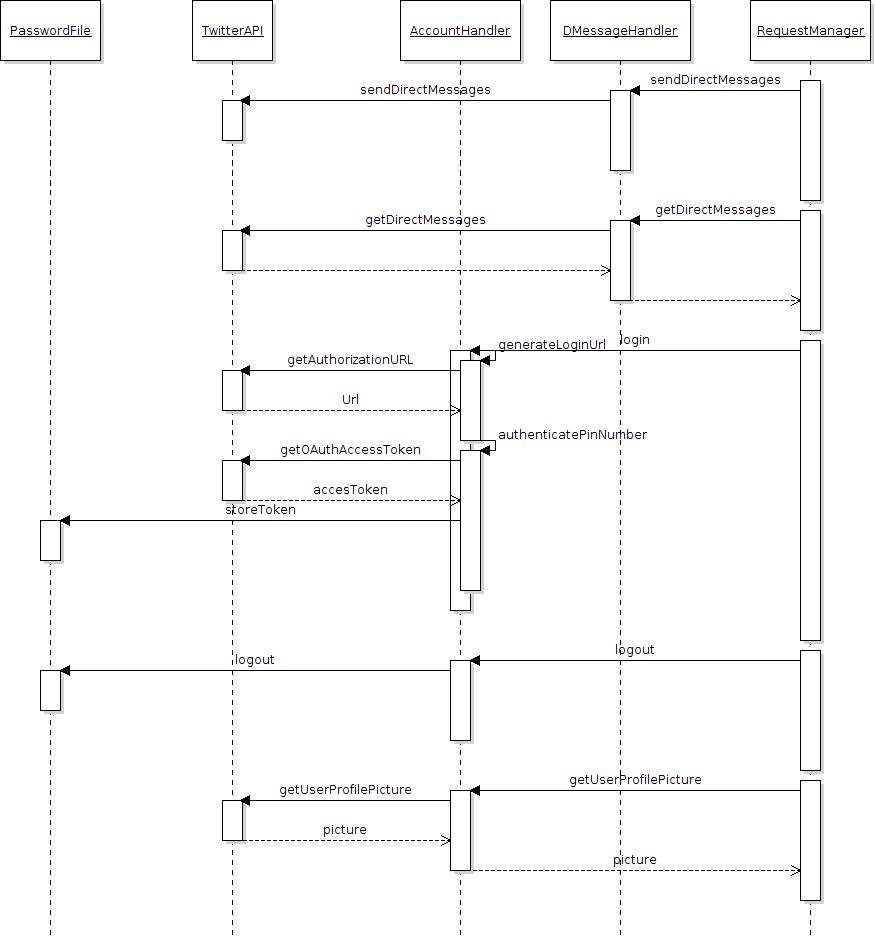
### Component Design Description

It will contain the classes to be used to communicate with the controller

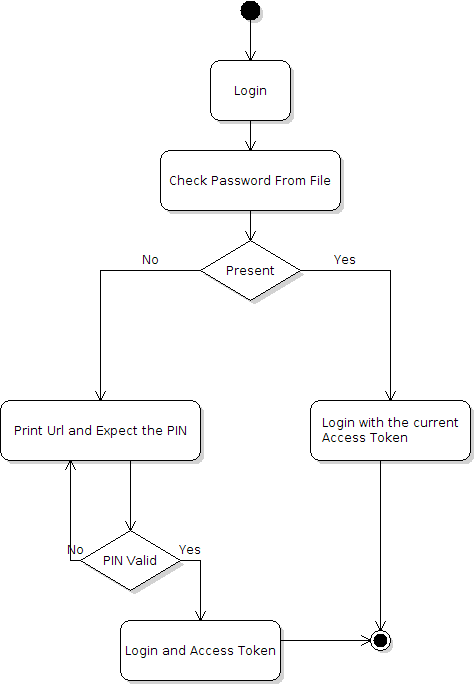


**Figure 3:** Handlers within Model

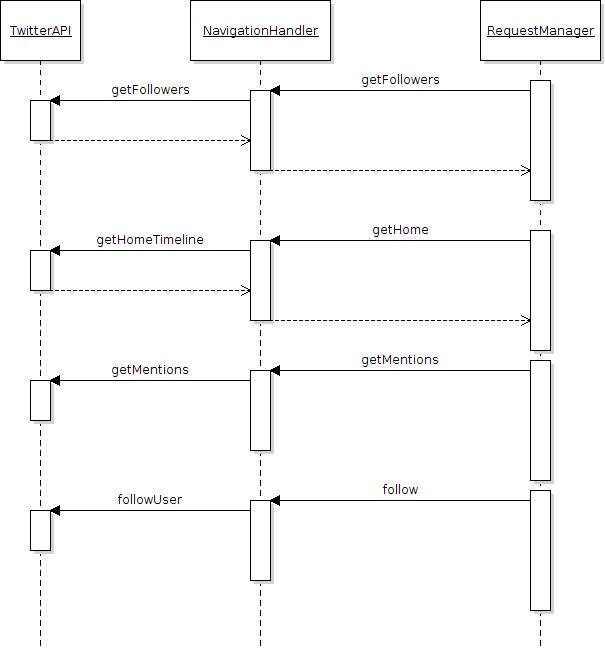
### Workflows and Algorithms



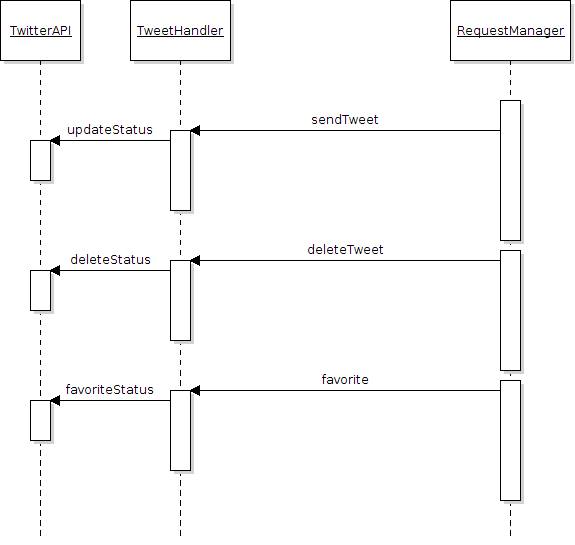
**Figure 4:** Sequence diagram between Model and Controller



**Figure 5:** Flow-chart for login process within Model



**Figure 6:** Sequence diagram between Model and Controller



**Figure 7:** Sequence diagram between Controller and Model

### Software Requirements Mapping

* SRS-001 Login
* SRS-002 Timeline Display
* SRS-003 Mention Display
* SRS-004 Profile Display
* SRS-005 Search
* SRS-006 Tweet Options
* SRS-007 Profile Settings
* SRS-008 Post Tweet
* SRS-009 Delete Tweet
* SRS-010 Send Message
* SRS-011 Display Messages
* SRS-012 Follow
* SRS-013 Unfollow
* SRS-014 Delete Authentication

## GUIManager and RequestManager

### Component Interfaces

GUIManager is a component between the GUI and RequestManager. It gets signals from Action Listeners within the GUI and tells the RequestManager what to do. RequestManager is component between GUIManager and DataRequester. GUIManager tells DataRequester what to ask from DataRequester. Both RequestManager and GUIManager don’t have a specific interface; rather they carries messages between DataRequester and GUI.

### Component Design Description





**Figure 8:** UML of GUI Manager and RequestManager

### Workflows and Algorithms



**Figure 9:** General sequence diagram between three major components

### Software Requirements Mapping

* SRS-001 Login
* SRS-002 Timeline Display
* SRS-003 Mention Display
* SRS-004 Profile Display
* SRS-005 Search
* SRS-006 Tweet Options
* SRS-007 Profile Settings
* SRS-008 Post Tweet
* SRS-009 Delete Tweet
* SRS-010 Send Message
* SRS-011 Display Messages
* SRS-012 Follow
* SRS-013 Unfollow
* SRS-014 Delete Authentication
* SRS-TDS-015 Timeline refresh
* SRS-TDS-016 Direct Message Limit
* SRS-TDS-017 Tweet Post Limit
* SRS-TDS-018 Characters per tweet
* SRS-TDS-019 Wait time for the PIN input

## GUI Component

### Component Interfaces

**MainFrame**

MainFrame is the base container of the GUI. It contains a NavigationBar, a MainContent and a TweetBox.

* + **NavigationBar**

NavigationBar is located at the top of the MainFrame and it contains these components:

* + Home Button
  + Notifications Button
  + Me Button
  + Search Button/Box
  + DMessage Button
  + Settings Button

Also, it handles the actions performed on these components.

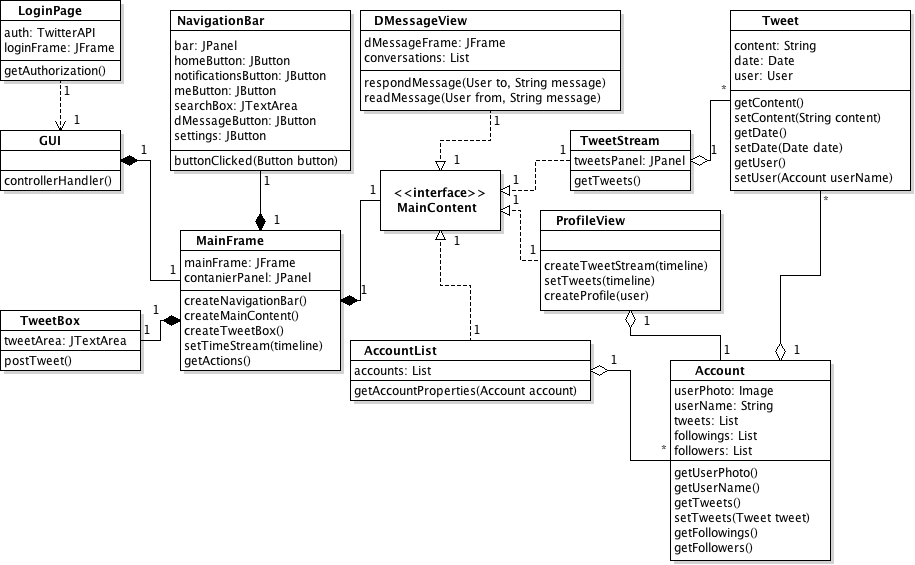
* + **MainContent**

MainContent is an interface for everything that can be shown in MainFrame. Followings are possible content types:

* + - **TweetStream** contains a stack of Tweets
    - **Profile** can be either user’s or others’
    - **DMessage** 
      * **DMessageList** is the list of previous conversations
      * **Conversation** is the current conversation
    - **AccountList**
      * **Followers**
      * **Followings**
  + **TweetBox**

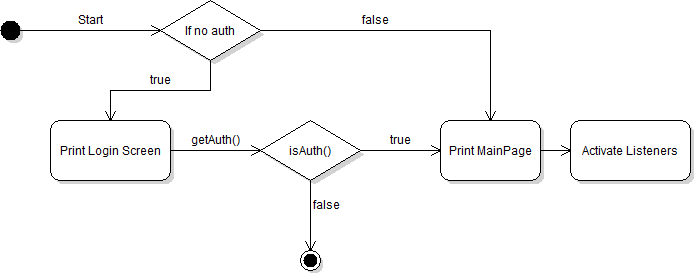
TweetBox is a place to write/post Tweets

### Component Design Description

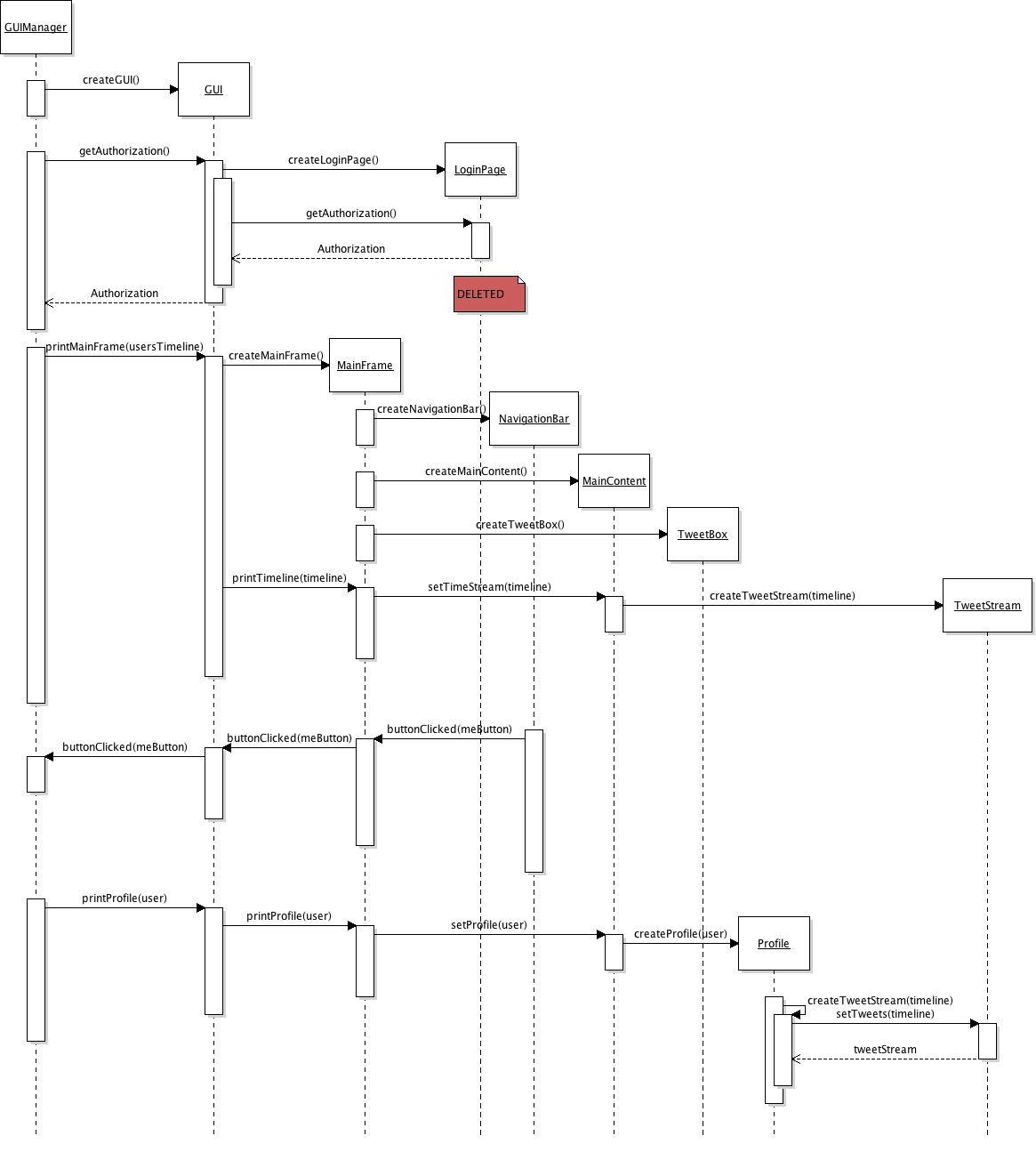


**Figure 10:** Class Diagrams for View Component

### Workflows and Algorithms



**Figure 11:** Start Diagram for View Component



**Figure 12:** Sequence Diagram for View

### Software requirements mapping

* SRS-002 Timeline Display
* SRS-003 Mention Display
* SRS-004 Profile Display
* SRS-005 Search
* SRS-006 Tweet Options
* SRS-007 Profile Settings
* SRS-008 Post Tweet
* SRS-009 Delete Tweet
* SRS-010 Send Message
* SRS-011 Display Messages
* SRS-TDS-015 Timeline refresh

# COTS Identification

COTS (commercial of the shelf) libraries used in TDS are the following:

* Twitter4j.jar, 3.05, <http://twitter4j.org/archive/twitter4j-3.0.5.zip> , Apache License v2.0