Software Requirements Specification

for

TA Processing System

Version 1.0 approved

Prepared by Garett Stage

UMN CSCI 5801

2/13/19

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document specifies the software requirements for the TA Processing System (TAPS). This document will cover the main architecture of the system, its use cases, and all features included.

1.2 Document Conventions

- TA – Teaching Assistant
- Prospective TA – Any student (graduate or undergraduate) who wishes to become a TA for the target term.
- Appointed TA – Any student who has applied and was granted a TA position in a given class.
- UMN – University of Minnesota Twin Cities
- Italics are used to distinguish between **main classes** and *subclasses* of users (i.e. **Administrators** including *administrative staff* and *payroll staff*)
    - Bold words indicate the main class.

1.3 Intended Audience and Reading Suggestions

This document is aimed at less experienced developers as a way to provide guidance on how to construct this system. Useful information can also be found that may be used by staff and managers to better understand the system in order to answer user-related questions. Begin by understanding the scope and conventions that this document takes before moving on to the specifications.

1.4 Product Scope

- To provide information on constructing this system to low experience programmers.
- To provide a system in which each essential user's needs can be satisfied
    - **Administrators** can perform data operations and administrative tasks pertaining to **prospective** and **appointed TAs**.
    - **Appointed TAs** have access to the information for the position in which they are appointed, and have control over the offer they received.
    - **Prospective TAs** have access to information regarding appointments. A place in which to submit academic, preferential, language, and technical qualifications as well as personal details required for appointment.
- Fulfill the budget, scheduling, and TA requirements set by the university for TA appointments.

1.5 References

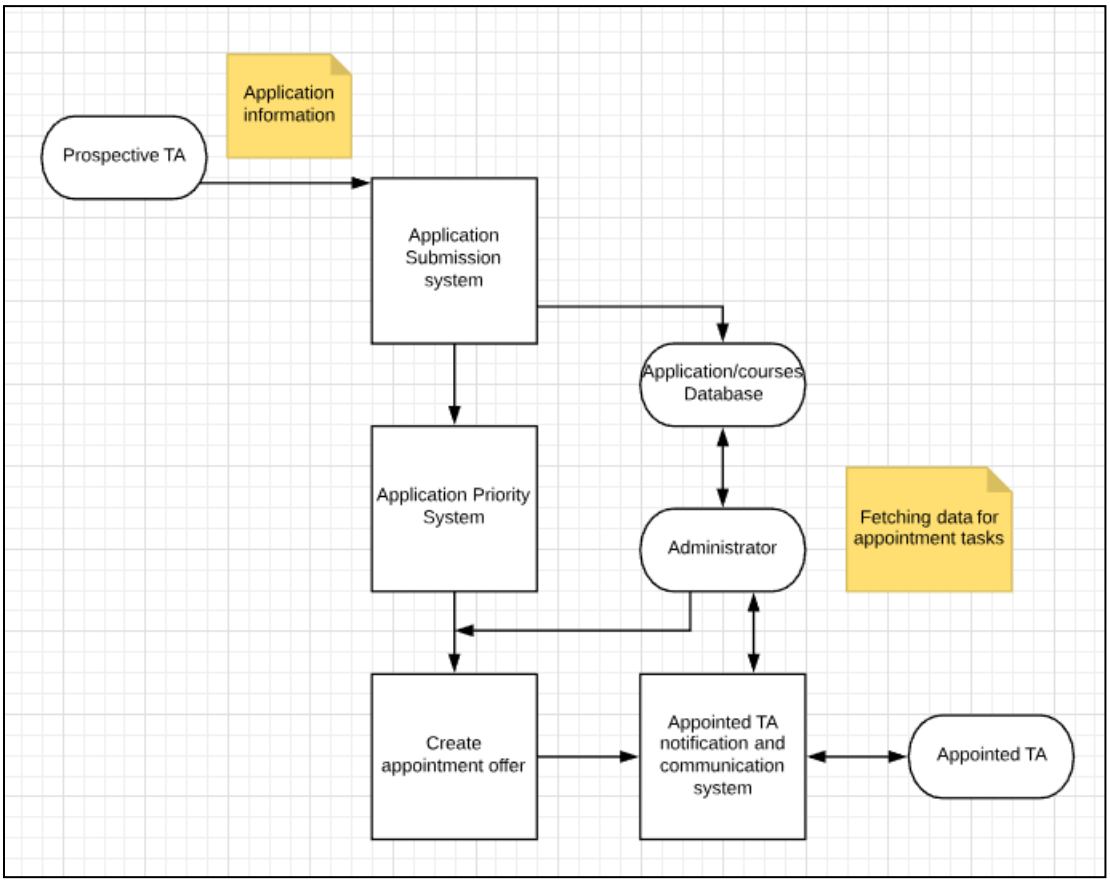
1.5.1) https://www.cs.umn.edu/academics/graduate/ta_handbook
1.5.2) https://www.cs.umn.edu/academics/undergraduate/engagement/ta-info

2. Overall Description

2.1 Product Perspective

This product aims to replace the existing system in place for appointing TAs. The goal is to migrate the original system, consisting of many different technologies, into a new, self-contained system. This allows for all of the information available to the administrators will be held in one place. Other technologies (Google Sheets/Mail) may be used, but must be streamlined to perform in a linear fashion. The diagram below illustrates the abstract subsystems contained within TAPS.

A description of the subsystems follows:



1. Application Submission
    a. This system allows the **prospective TA** user class to fill out a form of information detailed in this document to be processed by subsequent systems.
2. Application Priority
    a. This system entails computing a priority value for the information entered into the previous application submission system. This priority value is a measure of the importance for the applicant across all class offerings to be used by the administrator.

        i. The "Create appointment offer" is not a complete system, but a process defined to allow the system creator to understand the entire process. This process combines input from the administrator in the form of data analysis and input from the application submission and application priority systems in the form of recommendations (using priority values).
3. Communication system
    a. This system is a peer-to-peer communication that consists of multiple parts:
        i. User classes must be differentiated to users of the system. This allows each user to have specific privileges and proper information on who is communicating to who.
        ii. A notification system is to be implemented. This is to alert the **Administrator** user class to be promptly alerted to new events that take place within the TAPS system. These events include acceptance and rejection by **appointed TAs**, appointment updates by **Administrators**, and payroll updates by *Payroll staff*.

2.2 Product Functions

There are 3 major classifications of users. These are: **administrators**, **appointed TAs**, and **prospective TAs**.

1. **Administrators** are further divided into *administrative staff* and *payroll staff.*
    a. *Administrative staff* must be able to:
        i. View and calculate budget constraints
        ii. Fetch data for the list of courses being offered and estimate the number of TAs required.
        iii. Appoint TAs to courses based on priority score.
        iv. Update TA appointment status and communicate these to TAs, *administrative staff*, and *payroll staff*.
        v. Notify TAs with conditional offers (i.e. those that require language training).
        vi. Fetch and analyze data from previous years.
        vii. Receive notifications or communication about acceptance and rejection of offers and resignations.
    b. *Payroll staff* must be able to:
        i. Communicate appointment details to TAs.
        ii. Send and receive communication about appointment status updates.
2. **Prospective TAs** must be able to:
    a. Enter the following information to be submitted to administrators:
        i. Type of appointment to be requested and provide prior appointment details.
        ii. Course preferences and qualifications.
        iii. Additional qualifications (language skills, technical qualifications).
        iv. personal and academic details.
3. **Appointed TAs** must be able to:
    a. Accept or reject appointment offers.
    b. Communicate between **Administrators** for follow-up questions about the appointment.

2.3 User Classes and Characteristics

In order of priority:

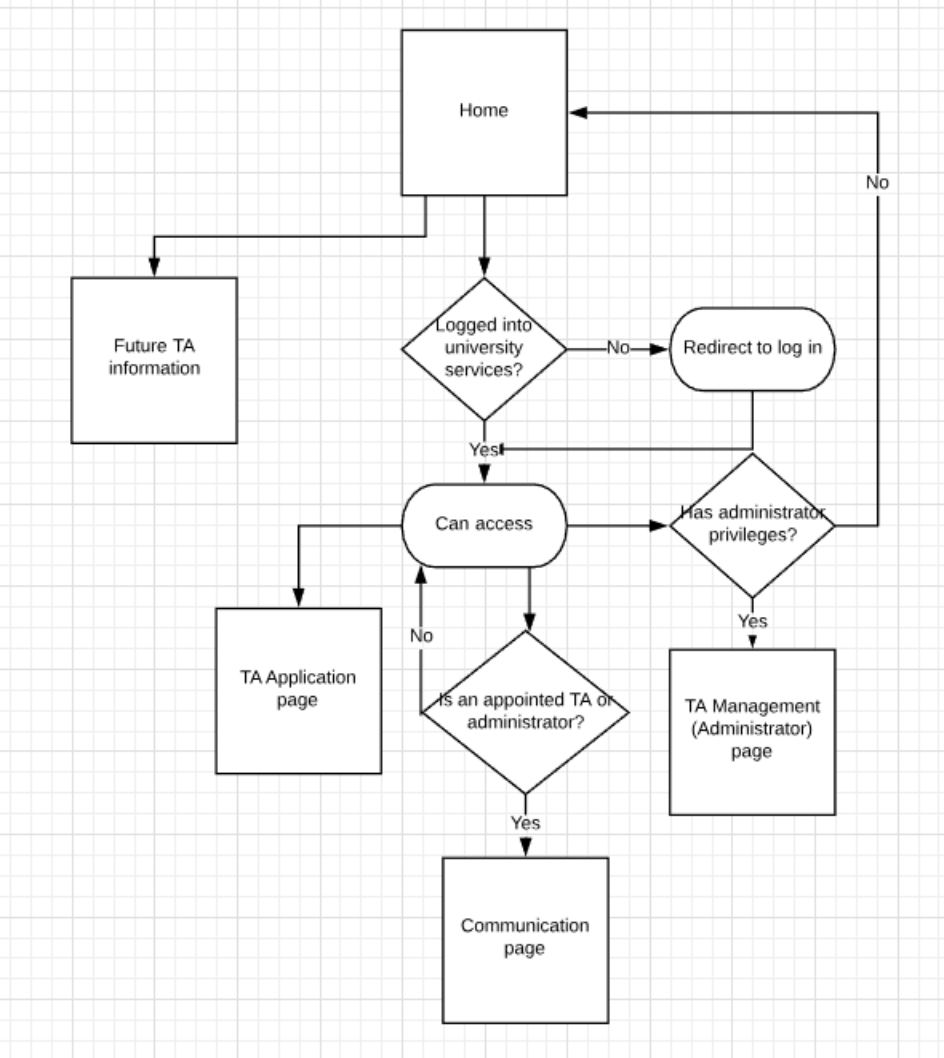
1. **Administrators**
    a. *Administrative Staff*
        i. *Administrative staff* make up the majority of interaction with this system. These users require access to all information and communication in the system, and will use each system in detail. Information from applications must be available to *administrative staff*, as this information will be used to choose appointments. This system aims to provide a streamlined way for this user class to create appointments as its main functionality. All aspects of the appointment system must give the ability for manipulation to this user class. This includes data access (both for classes offered and application data), and priority in the system for creating and updating appointments.
        ii. *Administrative staff* will use the communication system as well. The majority of communication is from *administrative staff* to/from *payroll staff*. All notifications sent to administrative staff pertains to updates in the *prospective* and *appointed TA* classes.
    b. *Payroll Staff*
        i. This user class is mainly responsible for communicating between the *appointed TA* class. This class uses the communication system for the majority of its interaction with the system. It answers questions and provides information to the *appointed TA* class. Open communication between these two classes is important.
        ii. This class requires access to the semester's class data and application data. This is because it will be used to decide information on budgets, as well as important information required by the *appointed TA* class.
2. **Prospective TA**
    a. This user class is responsible for the entering of information to the application database. 'entering' refers to forms submitted by the user through the application submission system to be entered into the application database. Required information is as follows:
        i. Appointment type and previous appointments
        ii. Course preferences and qualifications
        iii. Additional qualifications (Language, technical qualifications)
        iv. Personal and academic details
    b. This user class will be converted to the *Appointed TA* class upon receiving an appointment.
3. **Appointed TA**
    a. This user class is primarily responsible for communicating with the *Payroll staff* user class. This requires access to the communication system.

2.4 Operating Environment

The main operating environment will be using web environments and must work on all browsers. The goal of this system is to not require users to install the environment itself, and that it can be sustained in a browser environment. The top-level architecture is described in section (2.1) of this document. Network topology will be described in the External Interface (3.x) and (3.1) sections of this document. This will be operated on UMN servers, requiring access to the UMN log in services.

2.5 Design and Implementation Constraints

1. Appointments must be made in line with the University of Minnesota TA requirements (see (1.5.1) and (1.5.2) of this document.



2. Users who log in must have a privilege system according to their user class.
    a. **Administrators** have access to all functionality
    b. **Appointed TAs** have access to the communication system, but do not receive notifications on application submissions and updates to **Appointed TA** and **Prospective TA** status changes.
    c. **Prospective TAs** do not require a credential to access site features, as they do not have access to the administration page or the communication system.

3. External Interface Requirements

3.1 User Interfaces

The user interface relies upon the use of forms and information presented. Each page must present a specific set of information.

Below is the site topology and a description of each page and the information to be presented.

1. **Home**
    a. This is the landing page. The home page features a description of the site pages with available graphics to direct users to appropriate functional sites. Its purpose is to plainly describe the purpose of this site and the features therein. Access is granted to all users.
2. **Future TA Information**
    a. This page is meant to feature common questions and concerns placed by prospective TAs. It should contain links to (1.5.1) and (1.5.2) of this document and be updated to provide information that is commonly asked to **administrators**. Access is granted to all users.
3. **TA Application**
    a. This page contains the form that is available to applicants. This form contains options required in the **prospective TA** section (2.1) of this document. Submitting this form populates the application database with the information provided and notifies **Administrators** using the communication system. In order to access this page, users must be logged into university services using the UMN log in credentials. All users with these credentials are given access.
4. **TA Management**
    a. This page serves as the API for **Administrators**. This page has access to all past and current applications by **Prospective TAs** as well as information regarding past and present **Appointed TAs**. This information is segmented by **Prospective TA** and **Appointed TA** user types, shown in lists with the user's id and name. This segmentation can be shown in the form of two separate lists, or segmented into separate child pages. Access to this page must be accompanied by a UMN log in credential and a separate admin credential, provided in this system.
        i. **Administrators** can click on any of these names to view information given by applicants (**Prospective TA**) in section (2.1) of this document, as well as information about past positions mentioned in the **Appointed TA** section (2.1).
    b. If the administrator chooses a name, an optional "appoint" button is given to allow for the administrator to appoint the applicant. This opens up a list of classes that can be chosen. Upon choosing a class, the database updates to provide information that the applicant is now an **Appointed TA** and the class is updated to reflect the new appointment.
    c. If the administrator chooses a name, an optional "Resigned" button if given to allow for the administrator to remove the **appointed TA** from the class list. This removes the **appointed TA** from the list of TAs for the class they are assigned and places a remark on the TA for future knowledge.

5. **Communication**
    a. This page places the communication system. This system is used to provide notifications and peer-to-peer communication akin to email. This page can be accessed by **Administrators** and **Appointed TAs**.

3.2 Hardware Interfaces

N/A or not relevant.

3.3 Software Interfaces

This system will require a database which includes tables for:

- Application information described in section (2.1)

- User credentials
- Class appointments
    - This should include all classes with appointments, as well as **appointed TAs** for that class.
    - A remarks field is included to allow administrators to write notes that will show up when viewing information.
- Communication system messaging. This includes all notifications and messages sent through this service. Required information for this system includes:
    - Time and date of information sent
    - Id of user sending and the user receiving
    - Text sent as the header of the message
    - Text sent as the body of the message

UMN holds a database of class information that will be needed for the **administrator** user class. Interfaces of systems is described below.

1. **Application Submission**
    a. The application submission will take the form input submitted by the user and store it in the application database with the information presented.
2. **Communication/Notification**
    a. The communication system will query the database for all messages and notifications with regard to the current user's credentials. It is used to form the communication board to the user.
3. **Administrator API**
    a. The administrator API will query the database for the information placed into the application table (as described in section (2.1)) as well as previous appointment information with regard to that applicant. This is used to form the administration page providing the **Administrator** user class easily readable information on **prospective TAs**.

4. Use Cases

Use Case Name: Submit Application
Iteration: Filled
Summary:

1. The user navigates to the application page.
2. The user fills in the form with the desired information.
3. The user hits the submit button on the page.
4. The form is cleansed for improper input (i.e. SQL injection).
5. The information is added to the application table of the database.
6. Notifications are sent to **Administrators.**

Exception Paths: If the form is submitted with improper information, the user will be asked to resubmit the form, going back to step 2.
Extension Points: If the database connection does not exist in step 3, the user will be alerted and told to try again.
Assumptions: The user wishes to apply for a TA position and has UMN credentials.
Precondition: The user is not a TA and connections to the database exist.
Postcondition: The information is entered into the application table and the user is given confirmation.
Author: Garett Stage

Date: 2/13/18

Use Case Name: Appoint TA
Iteration: Filled
Summary:
1. *Administrative staff* user navigates to administration page.
2. *Administrative staff* finds student to appoint.
3. *Administrative staff* clicks the students name.
4. *Administrative staff* clicks the assign option on opened window.
5. Class is chosen from the list that is opened.
6. A notification is sent to the student and *payroll staff*.

Exception Paths: None
Extension Points: If the class is full, the *administrative staff* will be provided a warning and be placed back to step 5.
Assumptions: The user being chosen has fulfilled the "Submit Application" use case.
Precondition: The user being chosen is a **prospective TA** and the user choosing is an *administrative staff*
Postcondition: The user chosen is an **appointed TA** and appears for the class chosen.
Author: Garett Stage
Date: 2/13/18

Use Case Name: User A messages user B
Iteration: Filled
Summary:
1. User A navigates to the messaging page.
2. User A clicks the "new message" button on the page.
3. User A enters the address of user B.
4. User A enters the subject of the message.
5. User A enters the body of the message.
6. User A clicks the "submit" button in the message interface.
7. User B is sent the message and a notification.

Exception Paths: User A enters the address of user B incorrectly, the user will be prompted to enter it again.
Extension Points: None
Assumptions: The user has logged in with his UMN credentials
Precondition: The user sending the message is an **appointed TA** or **administrator.**
Postcondition: User B has the message in their messaging board.
Author: Garett Stage
Date: 2/13/18

Use Case Name: An **appointed TA** resigns
Iteration: Filled
Summary:
1. **Administration** receives notification of resignation.
2. *Administrative staff* navigates to the management page.
3. *Administrative staff* clicks the student that has resigned.
4. *Administrative staff* clicks the 'resign' button on the opened window.

5. The **Appointed TA** is removed from the class list and credentials revoked.

Exception Paths: The student resigning is not an **appointed TA**, an alert appears and instructs the *administrative staff* to confirm they have chosen the right person.
Extension Points: None
Assumptions: The student resigning has followed the process to resign and wishes to.
Precondition: The student resigning is an **appointed TA**.
Postcondition: The student resigning is removed from the class list and no longer has access to the message board.
Author: Garett Stage
Date: 2/13/18

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must perform queries without significant pause (no stalling) on Google Chrome, Internet Explorer 11+, and Firefox.

5.2 Safety Requirements

All data in transit and at rest must be encrypted. Standard encryption is AES-256, but SHA-256 is acceptable. All credentials must be in place to stop users from accessing portions of the site to which they should not have access. All forms, including the application form, must be cleansed to avoid SQL injection attacks.

5.3 Security Requirements

Appointments and resignations must correctly modify the permission levels of the user effected. At the end of each terms, the permission levels of each **Appointed TA** must be reset as they move out of the position. Access to user permissions is to be granted to IT personnel or **administration** members only.

5.4 Software Quality Attributes

This software must be adaptable to the needs of **administration**. This adaptability includes access and availability to information requested by these users that is not directly available through the site. At the request of **administration**, information not readily available through the site should be available to provide, with proper identification. Automatic notifications and messages must be modifiable at request or specifiable through the site itself.

5.5 Business Rules

To be determined