**MENTOR MANAGEMENT WEB APP**

**ARCHITECTURE DESCRIPTION**

**Chapter 1**

**Introduction**

The purpose of the requirements specification document is to gather together all the different aspects of the proposed Mentor Management System to give detailed in-sight into the requirements of the system. The document provides both a high-level overview of product features as well as defining the proposed system in depth.

**Overview**

The proposed system’s aim is to help university find suitable mentors who are either professionals in Gauteng or students. The system will operate as the Mentor Management System’s (MMS) digital replacement for current manual and unreliable system of operation.

There are two primary users of the system - the mentor and the mentee. There is one secondary user - the administrator of the mentor management system.

There are many benefits to the system the primary of which is that users do not have to physically locate and go to the administrator’s office to get or to be a mentor and that the administrator doesn’t have to manually match mentors to mentees.

Some of the secondary benefits that can be realized is the ability for mentors to keep track of the Mentors that register as well as coordinate the success programmes. To allow for mentees and mentors to edit their details and be matched with preferred mentors and mentees.

The risks that need to be taken into consideration are in level of High to Low. Some of the serious ones is information security and data loss. We also need to take a serious look at component damage over time and the influx of users at any time. This means the system needs to be able to handle multiple users at any given time without latency.

Successful and effective risk management is the basis of successful and effective IT security. Due to the reality of limited resources and nearly unlimited threats, a reasonable decision has been made concerning the allocation of resources to protect systems that will be introduced.

**Glossary**

**Mentor** – an experienced and trusted adviser

**Mentee** – a person who is advised by a mentor

**Pair** – A mentor and his/her mentee

**MMS** ­ ­- Mentor Management System; Proposed system.

**Mentor Management Scope**

This system is web application that will be able to register a mentor, requiring their demographic data, contact information, professional history to date, qualities of a preferred mentee and etc. System will also register a mentee requiring their demographic data, contact information, tertiary education details, brief description of themselves, preferred mentor and etc. This system will then match mentee with 1 or more mentors that more or less fits his/her criteria ordered by likelihood of preference. The mentee will then be able to “request” a mentor thereafter the mentor can then view the mentees profile which includes his/her contact details and can then choose to contact them with a response on whether or not they are willing to mentor the mentee. A mentor can choose to be inactive, not be visible in mentee’s “search results”, after they have found one or more mentees or at any given time. After a mentor registers to the system, an administrator will contact them and somehow verify the data given by the mentor before they are officially in the system. When a mentor chooses to mentor a student, they both have to set their status to paired and the student will be able to rate the mentor. This rating is viewed by the mentees when viewing a profile. A mentor can be a working individual and/or a student doing their 2nd year or up and the mentee is restricted to any university student.

This app is for university students and professional individuals. The functionalities will differ depending on who the user is, which is determined during registration. The system is facilitated by an administrator, who has access to all data stored in the system, verifies mentees and handles/records system reviews.

This system does not yet provide a platform for users to communicate.

**Chapter 2**

**Stakeholders and concerns**

**Mentors**

* Need accurate information about mentee; to accept a mentee they want to help.
* Needs to be able to opt-out of mentoring more mentees; if they can’t help anymore mentees.
* Needs to be able to deactivate their system account.
* Needs to see when their mentorship is requested; to know whose profile to view.
* Needs to be able to accept request; so mentee knows to contact them.
* Needs a user-friendly system.
* Efficiency
* Quick Responsiveness.

**Mentee**

* Needs accurate information about mentor; to make a better informed choice.
* Needs to find a mentor that matches their preferences; to get an effective mentor.
* Needs to know that the mentor is credible; for safety reasons and effectiveness.
* Needs to indicate interest in being mentored by particular mentor.
* Needs feedback on whether mentor is interested; to contact the mentor.
* Needs to be able to deactivate their system account.
* Needs a user-friendly system.
* Constant interaction and communication with their mentor.
* Efficiency.
* Quick responsiveness.
* Needs accurate storage of their personal data; mentor is matched more accurately.

**Administrator**

* View feedback about mentor; to know which mentors need help or whose accounts to deactivate.
* Notifications when a new mentor registers to system; so they can verify them.
* View mentor profiles; to use to verify the mentor.
* Needs to be able to deactivate users that are misusing the system
* Have to verify the mentor on the system; mentees need verified mentors.

**Use Cases**

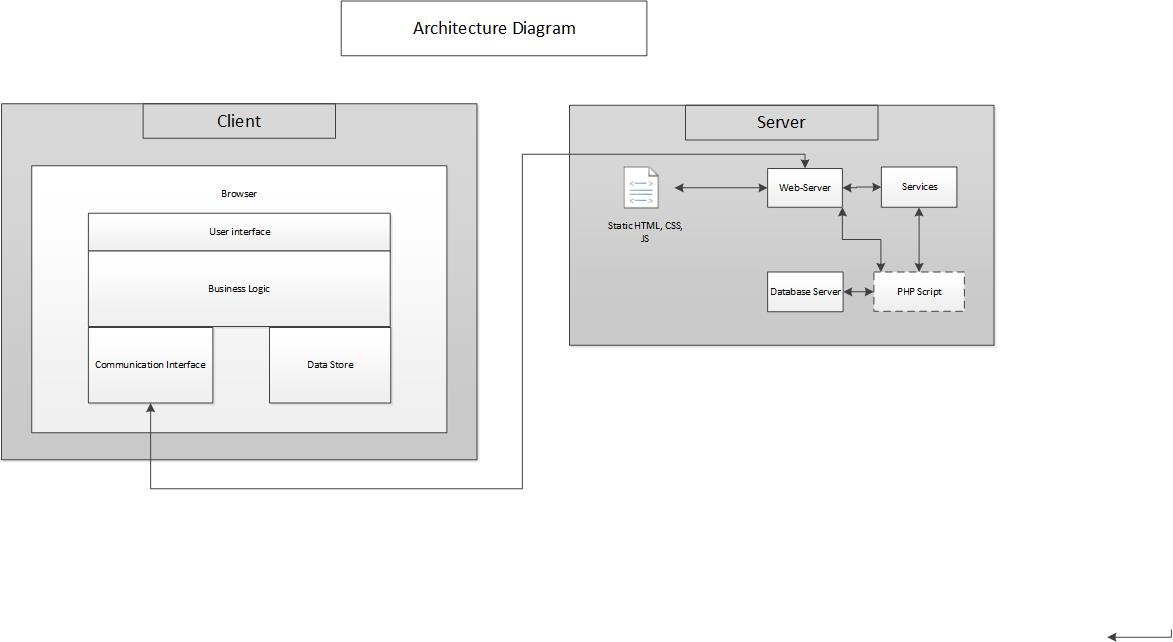
|  |  |  |
| --- | --- | --- |
| USER | FUNCTIONAL REQUIREMENTS | Use Case |
| General | System must allow user to register and specify the type of user (Mentor/Mentee). | Register User |
| System must allow user to login using a username and password | Login User |
| System must allow user to deactivate their account | Archive User |
| System must allow user to update their profile. | Update Profile |
| Mentee | System must allow mentee to fill out a form detailing preferred mentor. | Create Preferences List |
| System must allow mentee to update their mentor preferences. | Update Preferences List |
| System must allow mentee to search for a mentee; provide filtered suggestion based on preferences. | Search Mentor |
| System must allow mentee to view a mentor’s profile (including rating). | View Mentor Profile |
| System must allow mentee to request a mentor’s mentorship. | Request Mentor |
| System must allow mentee rate a mentor; provided the two are paired therefore if mentor has accepted the request. | Rate Mentor |
| \*System must allow mentee to mentor’s contact details; provided mentor has accepted the request. | \*View Mentor Contact Details |
| Mentor | System must allow mentor to receive request for mentorship. | View Request |
| System must allow mentor to view a mentee’s profile | View Mentee Profile |
| System must allow mentor to accept a request; pairing the two. | Accept Request |
| Administrator | System must allow admin to verify new mentors. | Update Mentor Verification |
| System must allow admin to view mentor ratings. | View Mentor Rating |
| System must allow admin to deactivate a mentor’s or mentee’s account. | Archive User |

**Chapter 3**

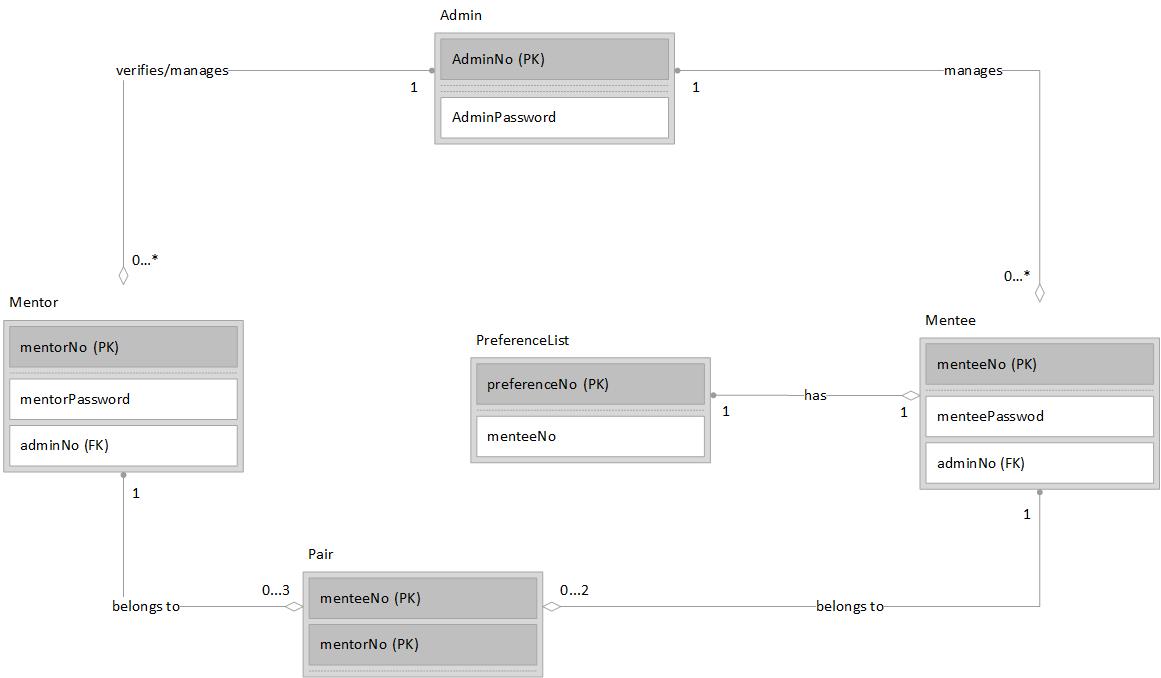
**Viewpoints+**

**Architecture Diagram**

This is a web application since it is easier to use on a personal computer and any mobile device which eliminates the responsibility of creating software for each platform but most importantly does not restrict any of our users to a certain type of device or operating system, making the system reliable.

****A layered architecture thus separating the presentation layer, the logic and the data layer from each other to implement a more test driven system by adding structure and ensuring that our “business rules” are not hard coded. Furthermore, usability is increased by the different UI standards.

**Class Diagram**

****This diagram depicts the entities and the relationship that they have with each other. It is high level overview of the classes the software has.

**Use Case Diagram**

As per functional requirements, this diagram depicts which use cases will be used by which user. These can be interpreted as methods and the users that those methods are implemented for.



**Main Functions Sequence Diagrams**

*Update Mentor Verification*

If a mentor has registered onto the system, they do not appear on mentee’s search list at all unless it has been verified somehow by the administrator that they are who they say they are on the system. A post condition for this use case is that the verified Boolean attribute for the mentor entity will be set to true, if the mentor is indeed credible.



*Create Preference List*

A precondition for creating a preference list is that the mentee must be registered on the system thus in the Mentee data store. And a post condition is that their preference list must be stored in the PreferenceList data store thereafter.

*Search Mentor*

A mentee’s search list consists of a list of mentors whose profile matches the preferences of that mentee by at least 4 characteristics. This list is ordered by the number of preferences matched with the highest number on top. A precondition for all listed mentors is that they have to be verified.

*Accept Request*

A precondition for this use case is that a request has to be sent first by the mentee. A post condition is that this mentor and mentee will now be paired thus included in the paired entity, allowing the mentee to be able to rate the mentor.