**3.4 Constraints**

The are multiple constraints that will determine how our team approaches the project. They have been divided into two categories, system constraints which are determined by the requirements of the finished system and development constraints which are factors which will affect the development process.

**3.4.1 System Constraints**

* **Accessibility** – The system should only be accessible by those who have permission and there must be a process for granting access to new users.
* **Usability** – The system should present the data in an easy and intuitive visual interface.
* **Data compatibility –** There is no current system in place however there is a large amount of existing data from different sources which the system must be capable of storing. The system must also be compatible with other data types which may be needed in the future.
* **Data management –** The system will need to allow for data to be created, updated, read and deleted.
* **Capacity –** The system will need the capacity to store various data dating back to 2009 and for 30-40 current users, the system will also need to plan for future growth.
* **Availability** – The system will need to be available at all times or as close to it as practically possible.
* **Accountability** – The system will need to log changes within the system so that any changes by a user must be accountable.
* **Redundancy** – The system will need to allow for changes to be undone if required or for the recovery of lost data.
* **Security** – There will be private and confidential information stored in the system, this must be secured as much as possible.
* **Maintenance** – The system will need to be designed so that it can be maintained by whoever has control of it in the future with as little technical knowledge as possible. If any changes need to be made to the system, it should be easy to determine how to effect those changes on the system.
* **Device compatibility** – There are no standard devices from which the system will be accessed from. The system will therefor need to be compatible with as many devices as possible and allow for a consistent experience across all of them.
* **Existing system compatibility –** It is not clear as of yet how the system will be deployed but it may need to be compatible with UWAs existing hosting solution or at least flexible.

**3.4.1 Development Constraints**

* **Timeframe –** The project will need to completed over one semester which will limit options, especially with areas of the project which will require additional learning. As it is being completed alongside other full-time study requirements, the number of available workhours will be limited and varying.
* **Existing knowledge** – Our team comes from a variety of different backgrounds and varying levels of familiarity with web development. We will have to carefully select frameworks and libraries in order to maximise use of our existing knowledge and cut down on learning time.
* **Project requirements** – In addition to the development of the product, the team will have to consider assessment requirements for the Professional Computing unit. Requirements such as work needing to be evidenced by GitHub commits will rule out certain development pathways.
* **Team** **member** **coordination** – It will be a challenge to coordinate work with team members who will have differing schedules and availabilities. Assigning roles and responsibilities so that everyone has the chance to contribute equally despite different areas of knowledge will be a challenge and these must also be rotated often. In addition, not all team members are located in Perth and having to coordinate work with remote team members will offer different challenges.
* **Budget** – There is no budget allocated to the development of the project so any option must be available without cost. It is also important to consider whether there are any costs associated with products when they are no longer used for personal or educational use.

**3.5 System Model**

**3.5.1 Scenarios**

There are many possible scenarios but the following have been selected to represent intended use and necessary requirements and presented in user-story format.

1. As a **potential user**, I want to have a clear pathway of gaining access to the repository so I can use its functions.
2. As a **potential** or **registered user**, I want only the required personal data to be stored and any confidential information that I input should be kept secure.
3. As a **registered user**, I want to be able to have a way to reset or change my login details in the scenario in which they have been forgotten or that I require for them to be changed.
4. As a **registered user**, I want to be able to upload, update, view or delete a file so that the information required is stored in the repository and it shouldn’t be a tedious process.
5. As a **registered user**, I want to be able to locate a required file quickly so I can perform the needed action on it.
6. As a **registered user**, I want the data to be sorted logically so that I can locate files related to what I require even though I may not know specifically what I need.
7. As an **admin user**, I want to control who has access to the repository so that data is only able to be accessed and modified by those who are authorised to.
8. As an **admin user**, I want the ability to restore or override data modifications and deletions in the event they were performed accidentally or maliciously.
9. As an **admin user**, I want the ability to monitor changes and track accountability so that in the event of an unintended use of the system I can prevent it or investigate what happened.
10. As an **admin user**, I want the ability to change parts of the system as they are needed and it shouldn’t be too complicated to do so.

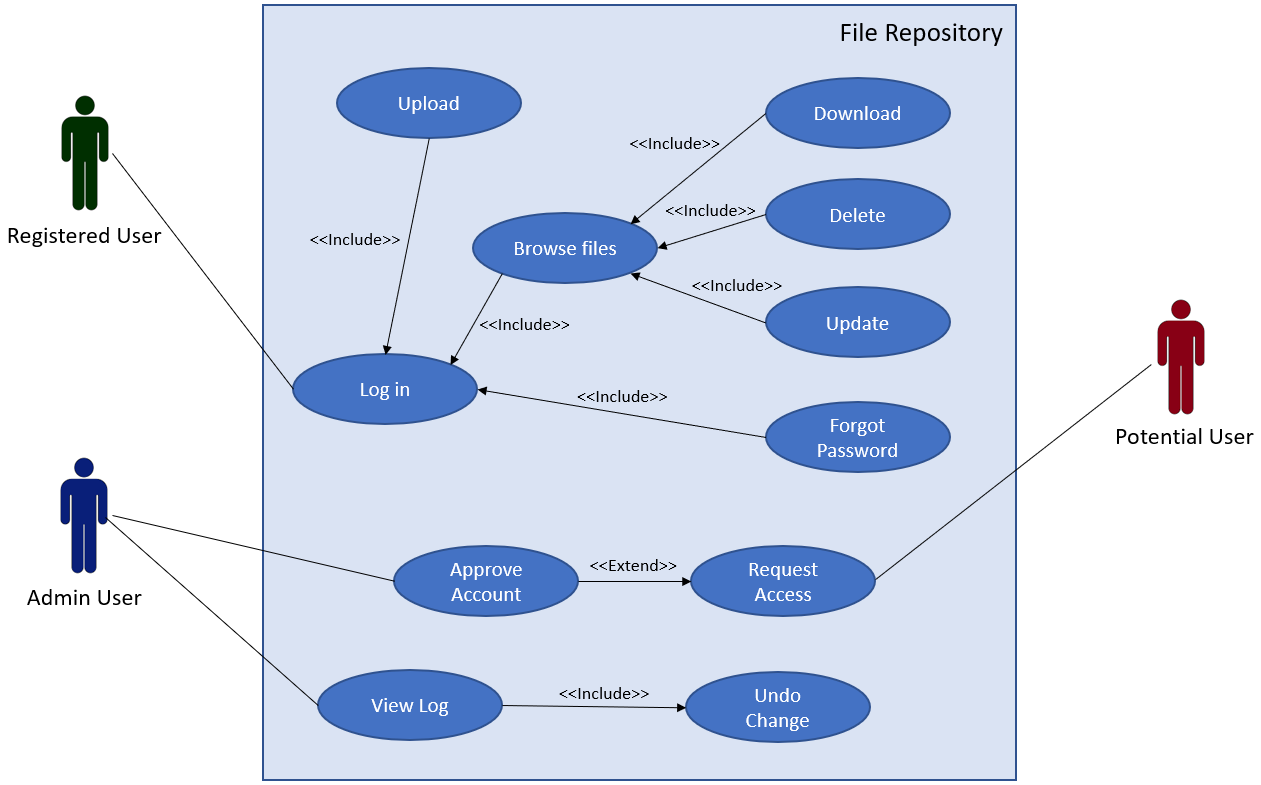
**3.5.2 Use Case Models**

**3.5.2.1 Actors**

There will be 3 types of actors who will interact with the system:

* **Potential Users** – Site visitors who have not yet been granted access to the repository.
* **Registered Users** – Approved users who have access to the repository.
* **Admin Users** – Users who have elevated permissions, who control access.

**3.5.2.2 Use Cases**

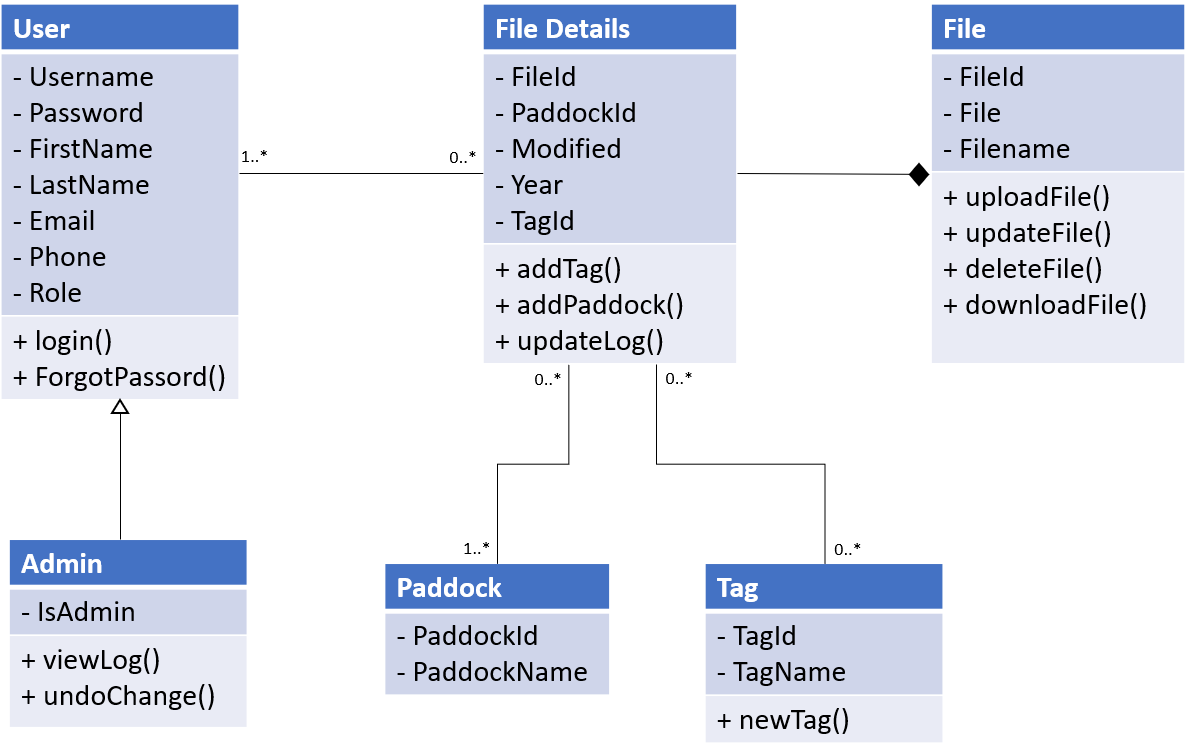


**3.5.3 Object Models**

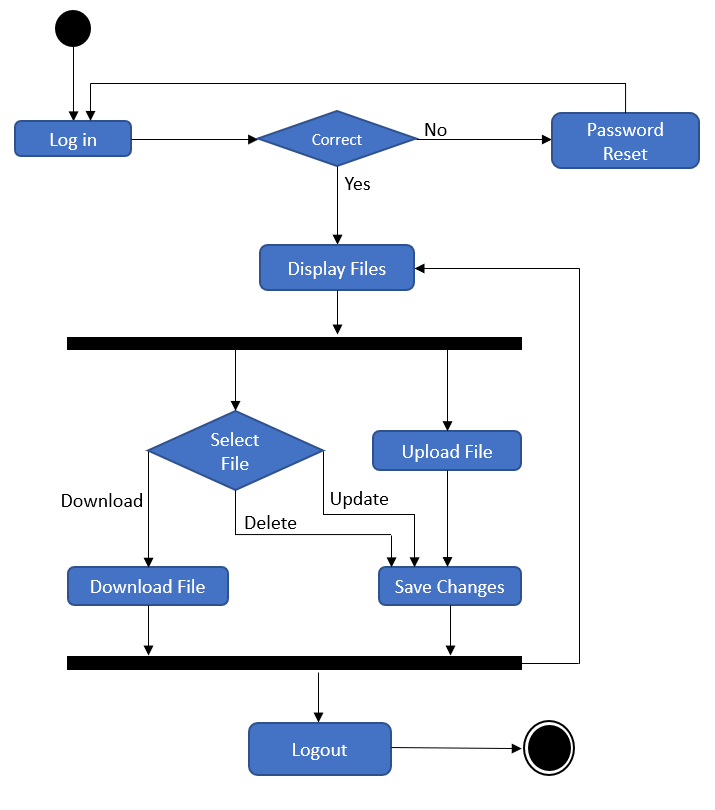
**3.5.3.1 Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Size** | **Description** | **Example** |
| UserID | Integer |  | Identifier for each user | 1 |
| FirstName | Text | 20 | First name of user | John |
| LastName | Text | 20 | Last name of user | Smith |
| Email | Text | 30 | Email address of user | jsmith@email.com |
| Phone | Text | 10 | Contact number of user | 0123456789 |
| Username | Text | 20 | Username for logging in | jsmith |
| Password | Text | 20 | Password of user, stored as hash value | 5e88489… |
| Role | Text | 20 | Role of user at UWA/Ridgefield | Researcher |
| Admin | Boolean | 1 | Whether user has admin permissions | False |
| FileID | Integer |  | Identifier for each file | 2 |
| File | BLOB |  | Actual file stored |  |
| Filename | Text | 20 | Name of the file | Example.docx |
| Modified | Timestamp |  | When file was last modified | 14/08/2022 19:00 |
| PaddockId | Integer |  | Paddock identifier | 1 |
| PaddockName | Text | 20 | Name of paddock | Paddock1 |
| Year | Integer |  | Year that file is associated with | 2022 |
| TagId | Integer |  | Tag identifier | 1 |
| TagName | Text | 20 | Keyword which can be associated with file | Soil |

**3.5.3.2 Class Diagrams**

****

**3.5.4 Dynamic Models**

**3.5.4.1 Activity Diagram**

**3.5.5 User Interface**

