# Introduction to Analysis

Analysis can be referred to as the process of defining what the user wants and expects from a system. After gathering enough information on project requirements, the specified requirements should be analyzed, well-documented and validated. The utmost need for analysis is to identify the needs of respective stakeholders and align them as per the need of business.

# Analysis Methodology

# Feasibility Study

It is necessary to keep track of the analysis process as well as the feasibility of thought-out strategies. With strong reference to factors affecting the system development, the pros and cons has been weighted out with appropriate explanations.

## Technical feasibility

## Operational feasibility

## Schedule

## Legal and social feasibility

## Economic feasibility

# Software Requirement Specification

## Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req.ID | Title | Description | Purpose | Dependences |
| FR1 | User Authentication | Assigning privileges as per the user’s authority | Security and integrity against a particular user’s action. | **N/A** |
| FR2 | Add course | Addition of new course by the authority | Including new trends and practices | **FR1** |
| FR3 | Update course | Update the contents of an existing course | Keeping up with the latest contents. | **FR1, FR2** |
| FR4 | Delete course | Delete the contents of an existing course | Excluding unwanted additions | **FR1, FR2** |
| FR5 | Display available course | Exhibit existing courses to the user | Interact and assimilate contents of an existing course | **FR1, FR2** |
| FR6 | Post comments on course | Leave comments on an existing course | Comment about additional queries and answers. | **FR1, FR2** |
| FR7 | Update comment | Update an existing comment | Edit or alter existing comment | **FR1, FR2, FR3** |
| FR8 | Delete comment | Delete an existing comment | Delete unwanted comments | **FR1, FR2, FR3** |
| FR9 | View comment | Display and read comments on courses | Display and read comments | **FR1, FR2, FR3** |
| FR10 | Create user profile | Create new learner profile | To save the details and progress of user | **FR1** |
| FR11 | Delete user profile | Delete existing user | Delete existing user | **FR1, FR10** |
| FR12 | Edit user profile | Update an existing user profile | Edit or alter existing user details | **FR1, FR10** |
| FR13 | Display user details | Display selected user details | Display details of a user profile | **FR1, FR10** |
| FR14 | Mark course as read | Mark selected course as read | Mark selected course as read | **FR1, FR5** |
| FR15 | Mark course as unread | Mark selected course as unread | Mark selected course as unread | **FR1, FR5** |
| FR16 | Blog CMS | Perform CRUD operations on blog posts | Perform CRUD operations on blog posts | **FR1** |
| FR17 | Repot issues | Report issues related to course | Report issues related to course | **FR1, FR5** |
| FR18 | View reported issues | View issues related to course | View issues related to course | **FR1, FR17** |
| FR19 | Delete reported issues | Delete reported issues related to course | Delete reported issues related to course | **FR1, FR17** |
| FR20 | Feedback CMS | Perform CRUD operations on course feedbacks | Perform CRUD operations on course feedbacks | **FR1, FR5** |
| FR21 | Request courses | Learners can request for courses | Requesting for addition of courses if not available. | **FR1** |
| FR22 | View requested courses | Display requested courses by learners | Display requested courses by users | **FR1, FR21** |
| FR23 | Delete requested courses | Delete requested courses by users | Delete requested courses by users | **FR1, FR21** |

## Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| Req.ID | Category | Description |
| NFR1 | Modifiability |  |
| NFR2 | Interface |  |
| NFR3 | Usability |  |
| NFR4 | Flexibility |  |
| NFR5 | Interoperability |  |
| NFR6 | Security |  |
| NFR7 | Performance |  |
| NFR8 | Data integrity |  |
| NFR9 | Modularity |  |
| NFR10 | Scalability |  |
| NFR11 | Load and Concurrency |  |

## MoSCoW Prioritization

|  |  |  |  |
| --- | --- | --- | --- |
| Req.ID | Requirement | MoSCoW | Justification |
| FR1 | User Authentication |  |  |
| NFR1 | Modifiability |  |  |
| NFR2 | Interface |  |  |
| NFR3 | Usability |  |  |
| NFR4 | Flexibility |  |  |
| NFR5 | Interoperability |  |  |
| NFR6 | Security |  |  |
| NFR7 | Performance |  |  |
| NFR8 | Data integrity |  |  |
| NFR9 | Modularity |  |  |
| NFR10 | Scalability |  |  |
| NFR11 | Load and Concurrency |  |  |
| FR2 | Add course |  |  |
| FR3 | Update course |  |  |
| FR4 | Delete course |  |  |
| FR5 | Display available course |  |  |
| FR6 | Post comments on course |  |  |
| FR7 | Update comment |  |  |
| FR8 | Delete comment |  |  |
| FR9 | View comment |  |  |
| FR10 | Create user profile |  |  |
| FR11 | Delete user profile |  |  |
| FR12 | Edit user profile |  |  |
| FR13 | Display user details |  |  |
| FR14 | Mark course as read |  |  |
| FR15 | Mark course as unread |  |  |
| FR16 | Blog CMS |  |  |
| FR17 | Repot issues |  |  |
| FR18 | View reported issues |  |  |
| FR19 | Delete reported issues |  |  |
| FR20 | Feedback CMS |  |  |
| FR21 | Request courses |  |  |
| FR22 | View requested courses |  |  |
| FR23 | Delete requested courses |  |  |

# Use Case Diagram

## Login and Registration

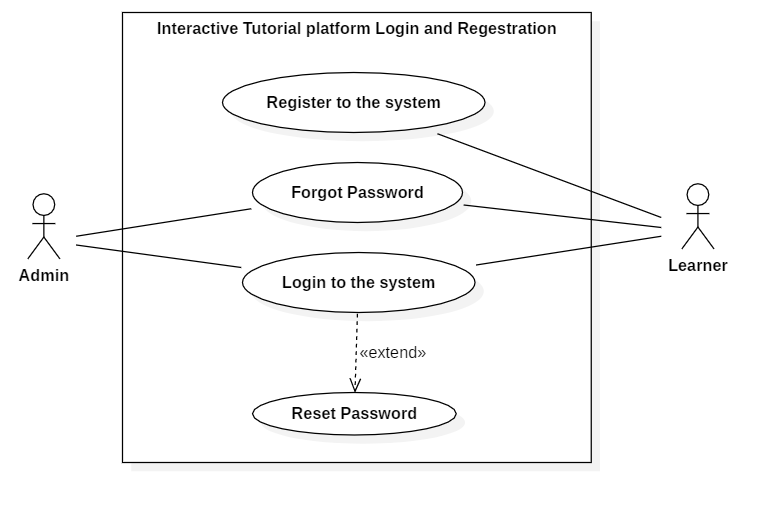


Figure : Login and registration

|  |  |
| --- | --- |
| **Justification** | The following use case depicts the flow of how a user logs or registers into the system viable for both admin and learners. |
| **Primary actor(s)** | Admin/Learners |
| **Secondary actors(s)** | *NA* |
| **Primary flow** | 1. Normal users initially register into the system leaving genuine profile information. 2. The user logs into the system via genuine profile information |
| **Alternative flow** | 1. In case the user forgets or loses their password, an option for resetting password is available. |

## Learner Profile

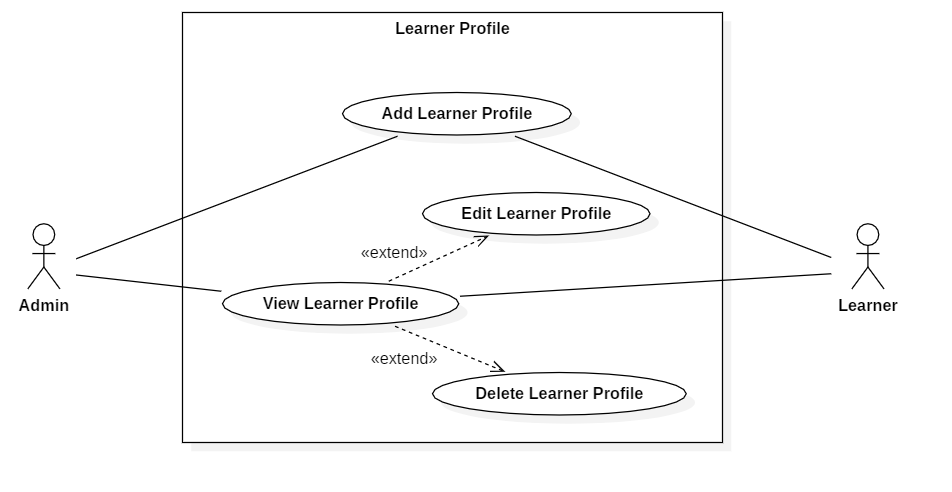


Figure : Learner Profile

|  |  |
| --- | --- |
| **Justification** | The following use case depicts the flow of how a user adds, views, edits and deletes learner profile. |
| **Primary actor(s)** | Admin/Learners |
| **Secondary actors(s)** | *NA* |
| **Primary flow** | 1. Normal users create a learner profile via registration (Admin user adds learner profile in some cases). 2. Both Admin and normal users can view learner profile with normal users having access only to their specific profile. 3. Once the profile has been viewed, an option for editing and deleting the profile credentials is available. |
| **Alternative flow** | 1. While adding or editing profile, users might input invalid data into the forms field. For this, via proper validations, users are to be asked to fill valid information. |

## Course – Admin

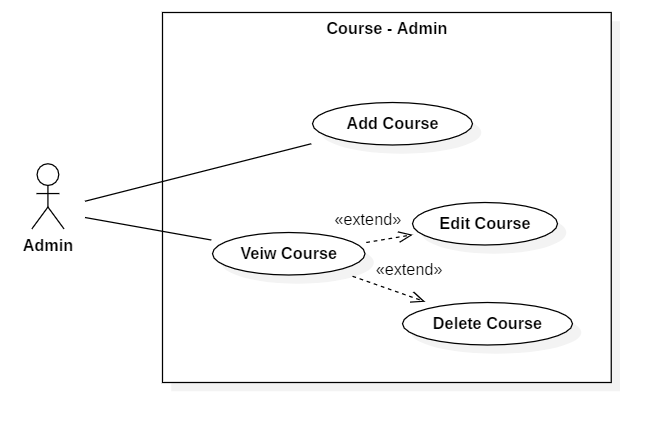


Figure : Course – Admin

|  |  |
| --- | --- |
| **Justification** | The following use case depicts the flow of how an admin user adds, views, edits and deletes a course. |
| **Primary actor(s)** | Admin |
| **Secondary actors(s)** | *NA* |
| **Primary flow** | 1. The admin user adds course to the system making it available for normal learner users. 2. Once the course has been added, the admin has the authority to edit or delete the content. |
| **Alternative flow** | 1. While adding or editing course, the admin might input invalid data into the field. For this, via proper validations, the admin is to be asked to fill valid information. |

## Course – Learner

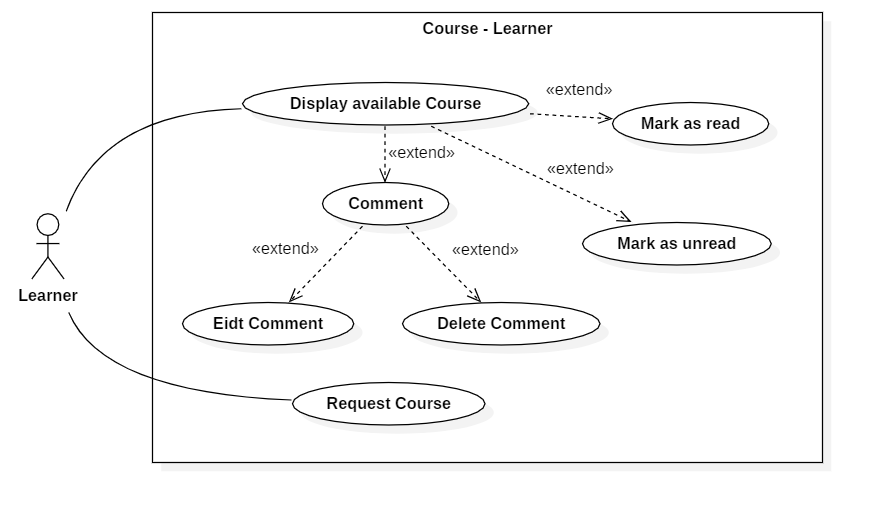


Figure : Course – Learner

|  |  |
| --- | --- |
| **Justification** | The following use case depicts the flow of how a normal user or learner accesses the available course. |
| **Primary actor(s)** | Learners |
| **Secondary actors(s)** | *NA* |
| **Primary flow** | 1. The user views one of the available courses. 2. The user has the privilege to leave comments on the course. 3. The comments left by a user can be edited and deleted by that particular user. 4. The courses, once completed or has fulfilled its purpose, can be marked as read or marked as unread at user’s will. |
| **Alternative flow** | 1. In case the user does not find what he/she is looking for, a field for requesting courses for future addition is available. |

## Blog

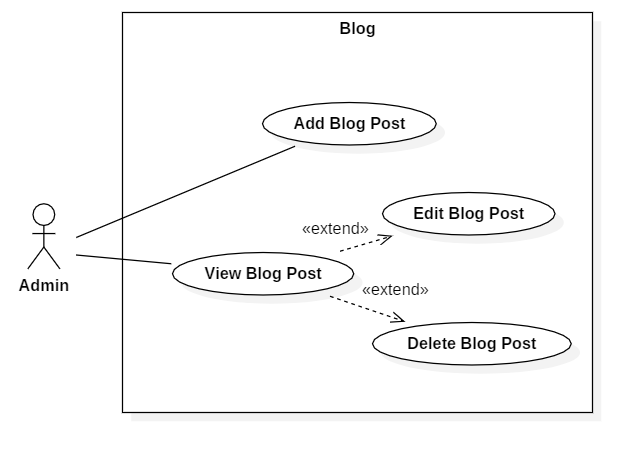


Figure : Blog use case

|  |  |
| --- | --- |
| **Justification** | The following use case depicts the flow of how an admin adds, views, edits and deletes blog posts. |
| **Primary actor(s)** | Admin |
| **Secondary actors(s)** | *NA* |
| **Primary flow** | 1. The admin adds blog post to the system dashboard making it possible for normal learners to view. 2. Once the blog post has been added, the admin has the authority to edit or delete an existing blog post. |
| **Alternative flow** | 1. While adding or editing blog post, the admin might input invalid data into the field. For this, via proper validations, the admin is to be asked to fill valid data. |

## Issues and Feedbacks

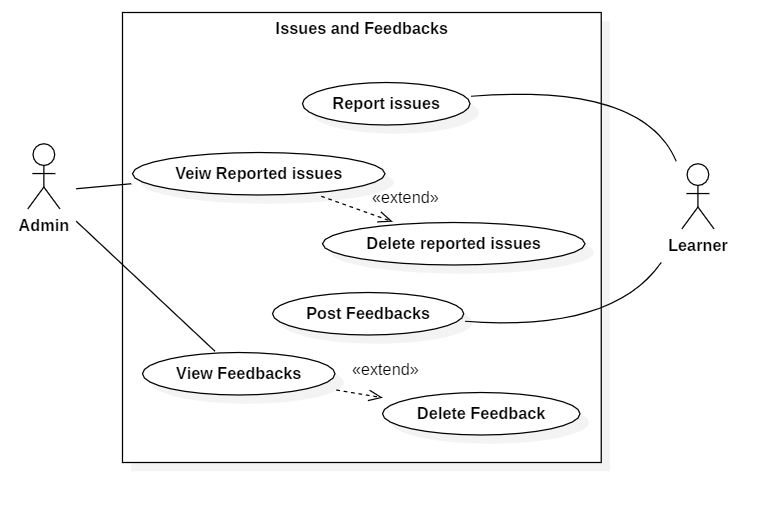


Figure : Issues and Feedbacks

|  |  |
| --- | --- |
| **Justification** | The following use case depicts the flow of how a user reports issues and post feedbacks for the admin to view and examine them. |
| **Primary actor(s)** | Admin/Learners |
| **Secondary actors(s)** | *NA* |
| **Primary flow** | 1. Normal users initially report some issues or bugs and post feedbacks for the admin to view. 2. The admin can view reported issues and posted feedbacks from normal learner users. 3. The viewed issues and feedbacks can be deleted upon rectification. |
| **Alternative flow** | 1. While reporting or posting issues and feedbacks, the user might input invalid data into the field. For this, via proper validations, the user is to be asked to fill valid information. |

# Initial Class Diagram via NLA

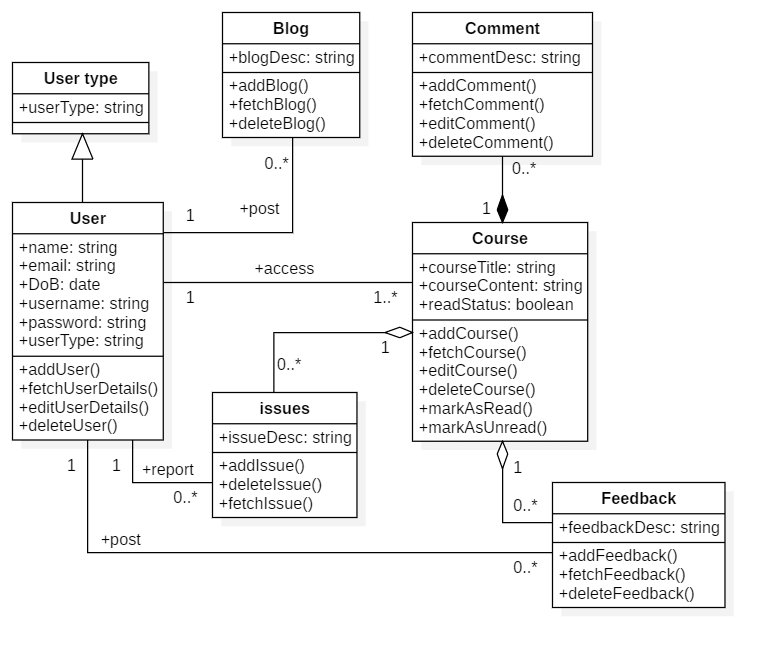


Figure : Initial Class diagram