Software Requirements Specifications

Daksh Shah Ananya Madireddy Tejas Cavale Aditya Mishra

Project : Recommendation Engine (Team 6)
Client : Dheeraj sir @ GradWise
17^{th} February, 2024

Brief Problem Statement

We are supposed to make a *Recommendation Engine* for GradWise which is a startup company focused on creating a novel learner-centric model for a learning platform.
This recommendation engine will suggest the next step that has to be taken by the users, depending on the previous performance (like tests taken before and many other factors).
These suggestions include:

- revising a previous topic
- giving a test for the current topic
- starting to learn the next topic

The project scope also involves attaching various tags for the questions(which can be of any form of media i.e. text, video, bot exercise) in the practice or quiz sets. This will make the preparation of quiz much easier and simpler, and less restricted. The project focuses mainly on course progress, current skill levels for various topics, learning methodologies modified with different type of courses. And any possible changes in the course structure should be handled.

System Requirements

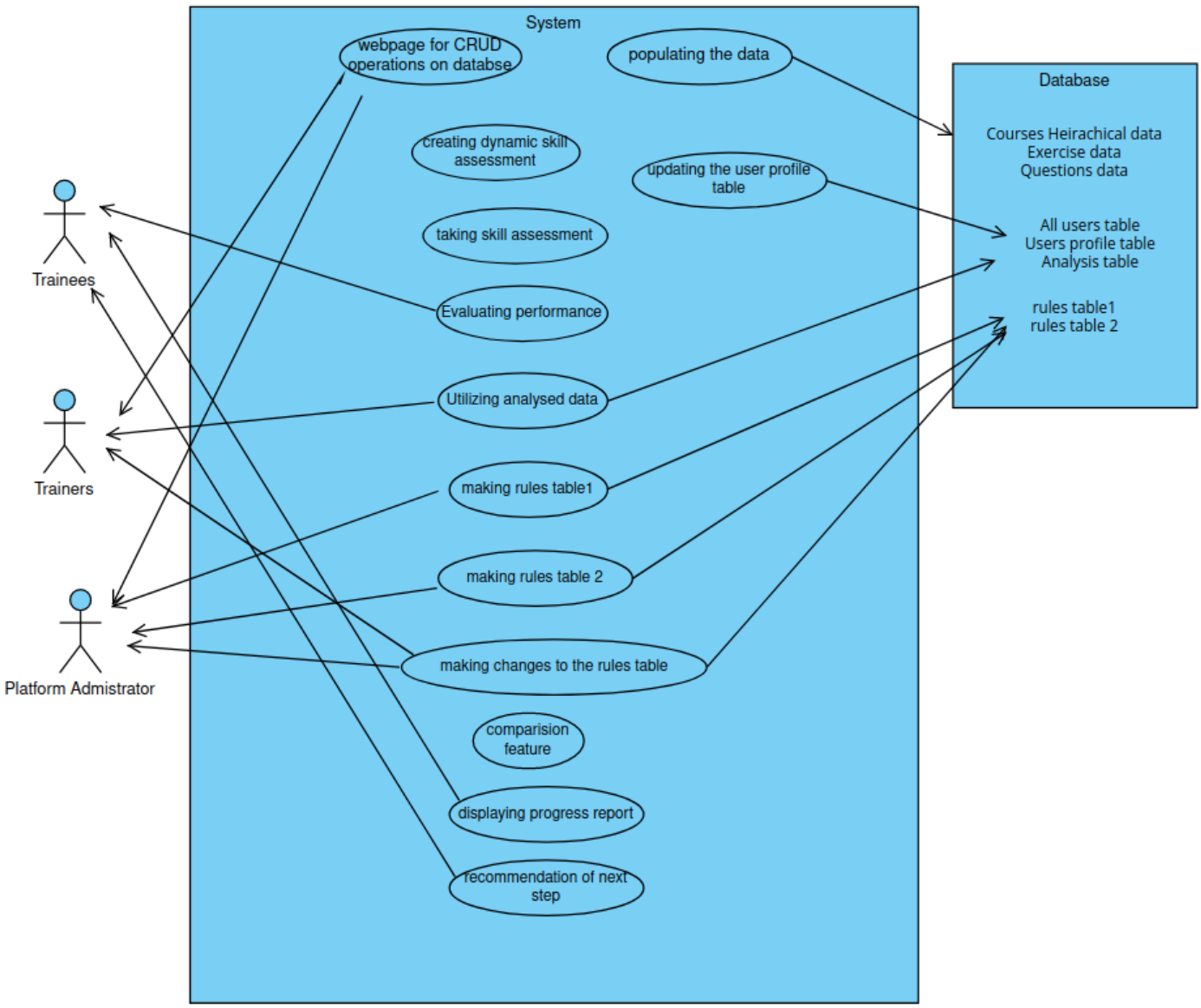
We are working on the environments of Node.js, React, Mysql. These all must be downloaded prior. These must be downloaded into the production environment. The project is bound to work on Firefox and Chrome as well.
Firefox Version 122.0
Google Chrome 120.0.6099.216 Technologies used:
1) Frontend: HTML,CSS, React
2) Middleware: Node.js
3) Backend: MYSQL

1 User Profiles

There are three users for the entire project:

- **Trainers:** There are the ones who design the course structure of the learning platform and make any possible changes, depending on the entire user community performance.
- **Trainees:** These are the people who belong to any educational institutions or personalised learning expected students, for a better progress and outcome of learning.
- **Platform admistrator:** A person who looks after every change, or the entire structure of the platform along with student progress. In other words, it could be the founder of the start-up.

Use-case Diagram



PHASE 1

Feature Requirements

This phase includes the below features:

- **Creation of Database schema;** database schema to store all the details of the courses, exercise data, along with the tagging system is to be implemented in tables with interconnecting tables along wise. We have used MySQL for the creation of database
- **Project Creation;** a sample project is created with API and UI for crud operations on the database. This will allow the system administrators and the trainers to change the course structure at any point of time. We have used Node.js, HTML & CSS for this
- **Creating content skeleton;** the entire project content has to be represented, using a simple data schema and sample test data. We have used React platform to create a sample project
- **Allowing MCQ and text base content;** while designing the exercise tables, there are mainly two types of contents present - MCQ's and text based formats. There can be other formats as well, such as bot exercise, theory question, coding question etc.
- **Interlinking content;** all the tags and the content must be interlinked with each other. Mainly in the format as follows: Course $\rightarrow$ Module $\rightarrow$ Sub-skill $\rightarrow$ Topic. Each topic can have pre-requisites connected to them, which will be further used in tagging
- **Adding meta tags to the exercises;** all the exercises are mapped to tags of course, module, sub skill, and skill topic, level, type of exercise, content type. All these tags are extracted by joining many tables together in the database, and establishing a connections and mappings

No.	Usecase	Description	Release
1	Meta data of courses	Storing learning content in orderly fashion for easy flow of CRUD ops	R1
2	Exercises data	Designing structure to store all data of all practice material (easily alterable too)	R1
3	Exercise content	Should be able to fit any type of content including images, videos etc.	R1
4	Tagging	Meta data of questions to store and map all possible tags for any particular exercise	R1
5	API	Used for accessing DB and make any possible changes using Node.js	R1
6	UI (HTML page)	Making accessible website to perform all CRUD ops on back-end	R1
7	Populating the data	Populating all tables with exemplar data to see if all functionalities work	R1
8	CRUD ops	Add, Delete, Update, Read ops should be performed on DB content	R1

Usecase Descriptions

Usecase number	1
Usecase	Metadata
Overview	Storing learning content in an orderly fashion for easy flow of CRUD ops
Actors	Trainers of learning platform and platform Admin
Pre-conditions	1^{st} usecase so no pre-condition
Flow	First step after start
Post-conditions	After this step, we must create DB of exercises

Usecase number	2
Usecase	Exercises content
Overview	Designing structure to store all data of all practice material (easily alterable too)
Actors	Trainers of learning platform and platform Admin
Pre-conditions	To fill data of the learning material
Flow	To fill data types of the practice material
Post-conditions	After this step, we must make tables to store the exercise content

Usecase number	3
Usecase	Exercises data
Overview	Should be able to fit any type of content including images, videos etc.
Actors	Trainers of learning platform and platform Admin
Pre-conditions	To fill the data of what is the practice material
Flow	To fill the data or content to be shown when the question is seen by the user
Post-conditions	After this step, we must make the tag all the exercises with their respective tags

Usecase number	4
Usecase	Tagging
Overview	Meta data of questions to store and map all the possible tags for a particular exercise
Actors	Trainers of learning platform and platform Admin
Pre-conditions	To fill the data or content to be shown when the question is seen by the user
Flow	To map each question to their respective tags in meta data
Post-conditions	Make API which accesses DB

Usecase number	5
Usecase	API
Overview	After this step we must make an API which that access the database
Actors	Trainers of the learning platform and platform administrator
Pre-conditions	To map each question to their respective tags in meta data
Flow	Use all the above data, and perform operations on them
Post-conditions	Populating the table

Usecase number	6
Usecase	UI design
Overview	Making a accessible website to perform all the CRUD operations on backend
Actors	Trainers of the learning platform and platform administrator
Pre-conditions	Precondition is to fill the data of the learning material
Flow	To map each question to their respective tags in meta data
Post-conditions	Populating the table

Usecase number	7
Usecase	Populating the database
Overview	Populating all the table to see if all the functionalities are working fine
Actors	Trainers of the learning platform and platform administrator
Pre-conditions	Completing API and UI design making a feasibility to populate
Flow	Filling the respective data using the UI made
Post-conditions	Check the working of the complete design

Usecase number	8
Usecase	CRUD operations
Overview	Add, Delete, Update, Read operations on the database content
Actors	Trainers of the learning platform and platform administrator
Pre-conditions	All the tables must be populated
Flow	Check all the functionalities using test cases
Post-conditions	Go to phase 2 implementation

PHASE 2

Feature Requirements

This phase includes the below features:

- **Content selection;** trainers can choose specific learning modules, topics, or questions to include in assessments. This allows for creating customized assessments tailored to specific learning objectives or skill areas
- **Rules Table;** defines criteria for selecting and presenting content in assessments. Examples of rules:
  - Learner skill level : *choose content based on the learner's current skill level or target learning goals*
  - Assessment type : *adapt content selection to different assessment formats (ex. multiple choice, open-ended)*
  - Learning path : *select content aligned with a predefined learning path or curriculum for the learner*
- **Assessment generation;** based on chosen content and applied rules, the system automatically generates assessments for learners to decide on the upcoming study plan & flow
- **Learner takes assessment;** learners complete the generated assessments, answering questions and demonstrating their skills
- **Performance evaluation;** the system evaluates learner performance based on their answers and predefined scoring criteria
- **Assessment data storage;** results from skill assessments, including learner responses and performance scores, are stored in the database
- **Data utilization for recommendations;** the recommendation engine leverages assessment data to do the following :
  - Identify skill gaps: *analyze learner performance to pinpoint areas where they need further improvement*
  - Personalize learning paths: *based on identified skill gaps and defined rules, engine should recommend learning materials that address the individual needs and target specific skills*

To summarize, this phase focuses on building functionalities for creating dynamic skill assessments, delivering them to learners, and integrating the resulting data into the recommendation engine to personalize learning recommendations.

No.	Usecase	Description	Release
1	Creating Dynamic Skill Assessment	Trainer builds customized skill assessment for learners	R1
2	Taking Skill Assessment	Students attempt skill assessment created for them	R1
3	System Evaluates Performance	Evaluating performance of the student	R1
4	R.E. Utilizes Data	R.E. refers to Rules table & processes next recommendation	R1

Usecase Descriptions

Usecase number	1
Usecase	Creating Dynamic Skill Assessment
Overview	Trainer builds customized skill assessment for learners
Actors	Trainers of learning platform and platform Admin
Pre-conditions	Trainer is logged in and has access to assessment creation tools
Flow	We are going to organise the questions stored in the tables into the practice tests, or exams which the trainee is going to take.
Post-conditions	New dynamic skill assessment is created and stored in the system

Usecase number	2
Usecase	Taking Skill Assessment
Overview	Students attempt skill assessment created for them
Actors	Students undertaking the assessment
Pre-conditions	Learner is logged in and has access to assigned assessments
Flow	Students or teh Trainees get to attempt these tests and must click submit to generate the results.
Post-conditions	Learner's answers and performance are recorded in the system

Usecase number	3
Usecase	System Evaluates Performance
Overview	Evaluating performance of the student
Actors	System
Pre-conditions	Learner has submitted the assessment properly
Flow	A screen displaying the results will be shown to the trainee. It has the correct answer, wrong answer statistics, concept lost etc
Post-conditions	Learner's performance is evaluated and a score is assigned

Usecase number	4
Usecase	R.E. Utilizes Data
Overview	R.E. refers to Rules table & processes next recommendation
Actors	System
Pre-conditions	Assessment data is stored in the database
Flow	We are going to store this data in json format and send it to the backend to tore and analyse the results.
Post-conditions	R.E. generates personalized learning recommendations for each learner

PHASE 3 & 4 (combined)

Feature Requirements

In these 2 phases, we will be performing the assessment of the skills of the users using the dynamic statistics and results of the exercises completed. We are going to use that to determine the current position, and personalising the path accordingly.
For this we are going to make the rules table first.

No.	Use Case Name	Description	Release
1	Rules Table 2	For analysing current position of user and what we are supposed to recommend	R2
2	Doc of assessment details	If statistics of skill level on current topic or previous must be updated in user table	R2
3	Script for analysis	Making rules table to work on statistics obtained	R2
4	Comparison feature	Comparing assessment details with rules table to check next step	R2
5	Next step decision	Taken from set of next steps to be taken for a user	R2
6	Progress report	Report of weak areas, strengths & improvements required	R2
7	Updating Analysis Report	Updating the user's progress towards end of module	R2
8	UI for Changes	For Any changes in rules table created above	R2

Usecase Descriptions

Usecase number	1
Usecase	Rules table 2
Overview	For analysing current position of user and what we are supposed to recommend
Actors	Trainers
Pre-conditions	There has to be a clear content and level analysis of each and every course from the rules table 1
Flow	Start filling the satisfying conditions for each exercise that has been suggested and taken by user
Post-conditions	Understanding & processing analysis

Usecase number	2
Usecase	Doc of assessment details
Overview	If statistics of skill level on current topic or previous must be updated in user table
Actors	Trainers and System Admin
Pre-conditions	All the after exam result data has to be captured
Flow	Divided into various forms of analysis like graphs, pie charts, etc. Categorising into respective modules
Post-conditions	Make a connection between this data and the rules table

Usecase number	3
Usecase	Script for analysis
Overview	Making rules table to work on statistics obtained
Actors	Trainers and System Admin
Pre-conditions	Both the data must be obtained before
Flow	Write a script to connect the rules table and the analysis table together
Post-conditions	Implementing a comparison feature

Usecase number	4
Usecase	Comparison feature
Overview	Comparing assessment details with rules table to check next step
Actors	Trainers
Pre-conditions	Connected data, must be compared with the rule table conditions present
Flow	Strengths and weak areas. Measuring need for improvement in those areas
Post-conditions	Make list of all possible recommendations that can be given to user

Usecase number	5
Usecase	Next step decision
Overview	Taken from set of next steps to be taken for a user
Actors	Trainers
Pre-conditions	All the results are analysed till this point
Flow	Organising data into increasing order depending on statistics and scheduling next segment in learning
Post-conditions	Displaying the recommendation

Usecase number	6
Usecase	Progress report
Overview	A report of weaknesses, strengths, improvement areas
Actors	Trainees or the users of the platform
Pre-conditions	All the data must be organised in the backend to be displayed
Flow	Making a UI which mentions each weak area and recommended exercise or practice or learning material
Post-conditions	Making the user choose what he wanted to improve first out of all those provided

Usecase number	7
Usecase	Updation of the report analysis
Overview	It updates the user's progress towards the end of the module
Actors	Trainers and Trainees
Pre-conditions	Trainees should be displayed the report analysis
Flow	Update the database regarding the completion of certain skill topic
Post-conditions	Making UI for changes in rules

Usecase number	8
Usecase	UI for changes
Overview	Any changes in the rules table created above
Actors	Trainers and platform administrator
Pre-conditions	Everything is updated and is in its place
Flow	Any changes in rule table which might occur depending on overall analysis of all the trainees
Post-conditions	Ready to implement