Vision and business case: (High level goals and constraints)

GOALS:

- The system manages different word list for different users and generates quiz for the user to keep track of learning and improving vocabulary.
- User gets randomly selected words from the database according to the words per day limit specified by user.
- System will also display image related to the word with some brief description if the words belongs to anatomy category.
- If user is searching for some specific word, then system will show him suggestions in search bar for easy access.
- It is worth investing in this project from the learning point of view.
- The system will give user words to learn throughout the week and at the end of the week user will get a quiz based on previously learned words.

CONSTRAINTS

- The time boxing is the major factor to be considered before building the system. Tasks which are incomplete should be added to future requirements considering the due dates.
- The budget should be noted before getting any requirements or planning.
- As per the costs, tools should be used such as methods, theories, functions, some software and hardware.
- The team should possess the necessary skills required. If not, one needs to undergo the training to build the system which would even help with the fast-changing technologies.

Use case model: (Functional requirements (10% in details))

Use case is a model of the system's intended functions and its environment. The use model is used as an essential input to activities in analysis, design and test. As per the system, some major functional requirements identified includes:

- User must find description and Images related to whatever he or she search from anatomy.
- Historical data must be present in order to generate quiz from the words learned by user.
- Authentication is necessary to separate user data from other users.
- Quiz must contain the words which are already searched or studied by user.
- System must give unique words on daily basis, at least 5 words or capacity specified by the user.

Supplementary specification: (Key and mostly non-functional requirements with impact on the architecture)

Describes other requirements, mostly non-functional. During inception, it is useful to have some idea of the key non-functional requirements that have will have a major impact on the architecture.

Following are some non-functional requirements:

- System should login user in 1.5 seconds if user enters valid username and password.
- User must see persistent view of the website.
- All link must see the correct page.
- Automatic logout if system is idle for 10 minutes.
- Legal and regulatory requirements, and application standards.
- Quality attributes of the system to be built, including usability, reliability, performance, capacity, maintainable, manageable, security, data integrity and supportability requirements.
- Other requirements such as operating systems and environments, compatibility requirements, and design constraints.

Glossary:

The Glossary defines important terms used by the project.

There is one Glossary for the system that provides a consistent set of definitions to help avoid misunderstandings. Initially use this artifact to understand the terms that are specific to the project

- **Developers**, who make use of the terms specified in the Glossary when designing and developing classes, database tables, user-interfaces, and many more than that.
- **Analysts**, who use the Glossary to capture project-specific terms so they can clearly define business rules, and to ensure that requirement specifications make correct and consistent use of those terms
- **Course developers and technical writers**, who use the Glossary to construct training material and documentation using specified terminology.
- **Gitlab/Github**, is a service that organizations can use to provide internal management of git repositories.
- **Jira** is a project management tool to work in a group.
- **Eclipse** is a development tool.

Sr. No.	Variable	Type	Description
1	Word	String	Anatomy words
2	Description	String	Contains description about the anatomy word
3	Username	String	Contains Username
4	Password	String	Contains password
5	wordLearnCount	Int	Will count the number of words user searches throughout the week.
6	wordImage	Image	Will store images related words.

Risk List and Risk management plan:

- Managing database of users, word list, and quiz
- Broken Authentication and Session Management
- Budget risks
- Server failure
- Failure to Redirect to specified URL, and retrieve words from database, suggesting words if user make a typo.
- Integration of different databases
- Data should be protected which is the main risk in any business.

Some Management Plans:

- Automated scanning and detection tools can be used to detect a good number of the above web security risks, but it is important to note that no detection software is infallible.
- All vulnerabilities identified should be documented in your website security risks report provided to you, so that you can get your IT support team focused on making your website more secure.

Phase Plan and software development plan:

The Software Development Plan is a comprehensive, composite artifact that gathers all information required to manage the project. It encloses several artifacts developed during the Inception phase and is maintained throughout the project. Initial phase includes the requirement gathering and designing for the model. The system is created in multiple small iterations. Under each iteration, testing is done and if the client demands some changes, then it can be easily updated and planned in the next phase.

- Tools: Wordpress, Github, Jira, Eclipse
- People: Admin, Contributor, Author, Editor, another user
- Education: Basic knowledge of anatomy
- Other resources: System, Internet

Development Case: (Customization of UP steps and artifacts for the project. In the UP, one always customizes it for the project)

Development case: The purpose of the Development Case is to capture the step by step process for an organization or for an individual project.

Steps: -

1) We decided the topic for prototype and based on it we write all the artifacts in detail specific to our project.

2) Project manager created Jira account and added all team members to share the work on it.

3) Web admin created the website and he gave rights to edit the website to all other team members.

4) We added the artifacts on the website.

5) Project manager created Github account and added all other team members to share work,

6) We added Github link to the website.

7) We begin creating a test database and started working on building chatbot for website.

FURPS+

Functional:

The system is reliable and user friendly to use. Moreover, it is always available to the user to search anatomy related words, view description and images. Quiz must be available at the end of each week. System provides security and confidentiality to the user; it will only allow authorised user to login.

Usability:

The system is very simple to use and easily understandable, though there is a help section in the website for user to solve their confusions. Help section contains basic FAQs, and some terms, and we also provided contact for users.

Reliability:

The system is reliable as it allows only authorised user to login and use the system. Moreover, time by time we are taking backup of each and everything like user details, word study details to face the situation of failure. So, it is easy to recover from the failure.

Performance:

Our system is very quick to respond the user's action, in order to do that we specified some time constraints while building the system. For example, system should login user in 1.5 seconds if user enters valid username and password. So, throughput of our system is high, and our system is always available to user accept the situation if system undergoes the maintenance and it is hardly 1 or 2 hours per month.

Supportability:

Our system is supported and configured on all platforms, such as Android, Windows, Linux, whether it is mobile phone, laptop or tablets.

The "+" of the FURPS+ acronym allows us to specify constraints, including design, implementation, interface, operation, packaging, legal and physical constraints.

- Implementation constraints –
    - An implementation constraint puts limits on coding or construction - standards, platform, or implementation language.
    - Resource Limitations: Sometimes insufficient amount of money could not allow you to acquire the specific resources you need.
    - Language and Tools: The system shall be developed using open source tools with java programming language.
- Interface constraints
    - An interface constraint is a requirement to interact with an external item. When you develop within an enterprise, quite often you must interact with external systems
    - Interfaces to existing systems: System should interact among external system.
- Operation
    - The team of people, who designed and developed system can run the system.
    - One single person i.e. administrator can also run the system.
- Packaging
    - As discussed in Operation Constraints, the team who developed system or administrator installs it.
- Legal
    - Licensing: It has an owner and can be sold or licensed.
    - Liability issues: Software should be liable in a precise and unambiguous way and establish such liability in case of incident.
    - Licensing fees or liabilities incurred from using third-party components or algorithms: If you do not own the copyright, you need permission to copy, or use the software, even if you wrote the code (e.g., you cannot make a copy for your personal use).