Nabu

Requirements Specifications

1. User Stories

1.1 Description

For the purposes of driving productivity, the initial batch of stories is formatted similarly to a scrum sprint, courtesy of Prof. Baer’s template from the Software Project Moodle course.

Sprint 1

Starts	16.10.2025
Ends	06.11.2025
Overall Goal	Establish basic user functionality

User stories that are a part of the initial sprint:

ID	Description	Acceptance Criteria	Priority
0	I am a developer working on this project and I require a base upon which I can build by adding and refining features	Establish a working environment fit for all developers working on this project	must have
1	I am a teacher and want to pool supplemental lecture material into an online database my students can easily access for studying	User can create repository for storing information in a user-friendly format	must have
2	I am a teacher looking for an easier way of assigning and grading homework assignments	User can create tasks which can be graded and assigned automatically	must have
3	I am a student looking for a more fun way to study	User can interact with gamified puzzles with an educational undercurrent	must have
4	I am a teacher and want my students to be able to access study materials, but do not feel comfortable sharing them publicly	Add ability to restrict certain repositories’ access	must have
5	I am an independent learner and want to use this app to learn about my hobby	Add publicly accessible repositories	must have
6	I am a teacher but I also want to participate in publicly available independent learning modules	Do not discriminate privileges based on user profile, rather on repository	should have

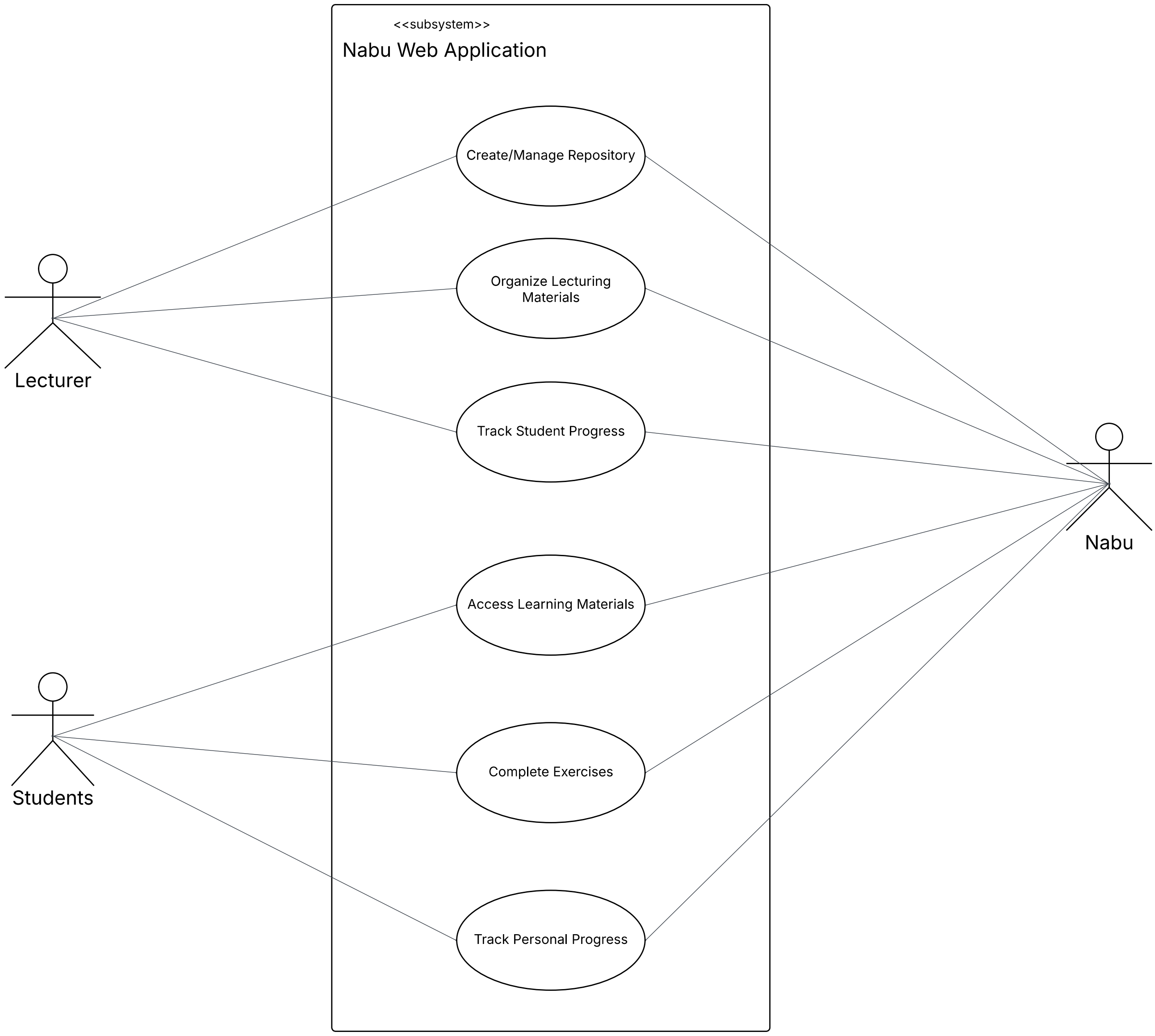
Priority enumeration: must have; should have; could have; N/A

1.2 Action/Mitigation

ID	Description	User Story ID	Effort (h)*		Status	Person(s)**
			Est.	Req.		
1	Establish a means of team communication and a unified working platform	0	1	1	done	David D. Radić
2	Find a productivity tool all team members can use uniformly	0	1	1	done	Benjamin Jäger
3	Compose a use-case diagram for the product	0	2	2	done	Khusee S. Bhingradiya Yashvi Patel
4	Determine appropriate technologies for building the product	0	5		done	Katherine J. Borromeo
5	Establish documentation structure	0	2		done	David D. Radić
6	Compose initial documentation	0	3		ongoing	David D. Radić Katherine J. Borromeo Benjamin Jäger
7	Establish a basic web infrastructure	0			ongoing	Katherine J. Borromeo additional TBD
8	Establish initial databases	0			ongoing	Katherine J. Borromeo additional TBD
9	Establish ability to create user profiles	1			ongoing	Katherine J. Borromeo additional TBD
10	User feature: create repositories for storing data	1			to-do	TBD
11	User feature: join repositories	1			to-do	TBD
12	User feature: invite to repository	1			to-do	TBD
13	User feature: create interactable puzzles	2			to-do	TBD
14	Feature: repository members assigned puzzles automatically	2			to-do	TBD
15	Feature: puzzles are graded automatically	2			to-do	TBD
16	User feature: complete puzzles	3			to-do	TBD
17	User feature: track puzzle progress	3			to-do	TBD
18	User feature: restrict repository access	4			to-do	TBD
19	User feature: add user to restricted repository (invite/accept)	4			to-do	TBD
20	Feature: add repositories which can be joined freely	5			to-do	TBD
21	Feature: users can both create and join repositories	6			to-do	TBD

*Values rounded up to the highest integer, empty fields are TBD
**Team member Tarek Alhidar has been recently unavailable due to illness
Status enumeration: done; ongoing; to-do; deferred; abandoned

2. Use Case Diagram



3. Constraints

Restriction	Description
Technical	
Technology - Languages	Javascript, HTML, CSS Additional scripting language(s) will be considered only if deemed absolutely necessary
Tech. - Frameworks	Node.js, React.js, npm Additional Javascript packages and frameworks will be considered on a use-case basis
Data Storage	SQL, MySQL
External Systems	No external APIs
Working Environment	GitHub
Development Environment	Advised but not mandated: Visual Studio Code with specified extensions for continuity, uniformity and compatibility
Task Allocation and Delivery	Scrum via Jira OpenUP via Prof. Baer’s Software Project module
Team Communication	Dedicated Discord server for project discussion
Product Security	Research into and seek compliance with relevant laws, regulations and international standards, especially regarding information security
Performance	Product must be reasonably fast and responsive
Product Scalability	Product includes ability for users to create abstract databases containing data.
Product Availability	As a public learning web app intended for independent learning the product should be available online at all times excluding maintenance and outages outside of the control of the development team
Documentation	Product development should be thoroughly
Non-Technical	
Compliance - Legal	Product and project must comply with relevant laws regarding administration, documentation, integrity and security
Comp. - Regulatory	Product and project must comply with regional regulations in all operative regions
Comp. - Best Practices	Product and project should comply with best practices regarding documentation and code infrastructure
Team Experience and Time Frame	Product is limited by the development team’s prior experience and the timeline specified by the Software Project module structure. Most team members are not broadly experienced with the used technologies
Team Availability	This project is a part of a college course and as such all team members are full-time students, upon which many also have part-time jobs. The expected allocated time is 10 hours per week per member on average as outlined by Prof. Baer

4. Risks

As permitted by prof. Baer Priority/Probability/Impact are TBD

ID	Topic	Description	Consequence
1	Authentication Data Integrity	Storing user IDs and passwords as raw text or inadequately encypted	User data leakage is trivialized Potentiates impersonation
2	User Data integrity	Storing user e-mails and usernames as raw text in the database or inadequately encrypted	User data leakage is trivialized Potentiates doxxing Potentiates spamming/phishing
3	Script Injection	Inserting malicious scripts via text input fields	Remote arbitrary code execution by malicious actors
4	SQL Injection	Leveraging text input fields to gain access to the SQL database	User data leakage via auditing the SQL database
5	OS Command Injection	Leveraging text input fields to execute OS-level commands on the server	Remote arbitrary code execution by malicious actors
6	DDoS	Overload the website infrastructure with dummy requests	Product is rendered inaccessible
7	Insecure APIs	External API used to build project is compromised and thus endangers the product	API vulnerabilities are inherited by the product
8	Social Engineering	Includes among others impersonation of users or administrators and phishing (sending malicious emails)	Elevation of privileges, data/identity theft
9	Cross-Site Scripting	Inserting malicious scripts into the web page to be executed by a user's browser	Remote arbitrary code execution by malicious actors
10	Insecure File Uploads	Using the repository feature to upload malicious files	Remote arbitrary code execution by malicious actors (in synergy with social engineering)
11	Actionable Media Uploads	Using the repository feature to upload harmful and/or illegal materials	Project team is potentially an accomplice in committing a crime

Priority/Probability/Impact enumeration: crit, high, med, low, none