

Project Report #1

Quiz of Fury

Daniel Seip
Daniel Terreros
Julia Rogers

Github URL: https://github.com/DanielTKC/quiz_of_fury/

CSCI 441

Table of Contents

| | |
|--|-----------|
| 1. Customer Problem Statement..... | 4 |
| a. Problem Statement..... | 4 |
| b. Decomposition into Sub-problems..... | 6 |
| c. Glossary of Terms..... | 7 |
| 2. Goals, Requirements, and Analysis..... | 8 |
| a. Business Goals..... | 8 |
| b. Enumerated Functional Requirements..... | 9 |
| c. Enumerated Nonfunctional Requirements..... | 10 |
| d. User Interface Requirements..... | 11 |
| 3. Use Cases..... | 14 |
| a. Stakeholders..... | 14 |
| b. Actors and Goals..... | 15 |
| c. Use Cases..... | 16 |
| i. Casual Description..... | 16 |
| ii. Use Case Diagram..... | 17 |
| iii. Traceability Matrix..... | 18 |
| iv. Fully-Dressed Descriptions..... | 19 |
| d. System Sequence Diagrams..... | 22 |
| UC-2: Study Flashcards..... | 22 |
| 4. User Interface Specification..... | 24 |
| a. Preliminary Design..... | 24 |
| UC-1: Create Flashcard Deck..... | 24 |
| UC-2: Study Flashcards..... | 27 |
| b. User Effort Estimation..... | 29 |
| UC-1: Create Deck with 10 Cards..... | 29 |
| UC-2: Study Session with 15 Cards..... | 30 |
| UC-3: View Dashboard..... | 30 |
| Project Management..... | 31 |
| Project Timeline Overview..... | 31 |
| Sprint Planning..... | 31 |
| Sprint 1: Foundation Setup (June 12-19, 2025)..... | 31 |
| Sprint 2: Core Functionality (June 19-26, 2025)..... | 32 |
| Sprint 3: Polish and Gamification (June 26-July 3, 2025)..... | 33 |
| Sprint 4: Testing and Demo Preparation (July 3-8, 2025)..... | 33 |
| Sprint 5: Final Polish and Presentation (July 8-18, 2025)..... | 34 |
| Team Responsibilities and Coordination..... | 35 |
| Role Definitions..... | 35 |
| Communication and Coordination..... | 35 |
| Risk Mitigation..... | 35 |

| | |
|------------------------|-----------|
| References..... | 36 |
|------------------------|-----------|

All Team Members Contributed Equally To This Report

Summary of Changes:

- Headings added to customer problem statement for clarity
- Project Part 1 d. User Interface Requirements has been combined with Part Two d. User interface specifications.
- User interface requirements have been updated to be more in line with their final appearance
- Updated Week one sprint status
- Spaced repetition algorithm implementation moved to stretch time due to complexity
- Added conclusion addressing pros, cons, cost, and expected investment return

1. Customer Problem Statement

a. Problem Statement:

i. The Reality of Student Life: Time Constraints and Academic Pressure

I am a full time college student with a full time course load. I also have a job that takes up a lot of my time outside of class, breaking up time that could traditionally be spent on only studying. I can typically only study at night or in 20 minute intervals between getting to various classes or breaks at work. There is an almost overwhelming amount of material to learn from all of my classes, sometimes with very little time to work on mastering it! I care deeply about learning this material - not just for the sake of a grade but so that I actually understand what I learn and can apply it after the class ends. I have tried different study methods: I wrote out notes, made flashcards by hand, re-read texts, even used current flashcard type webapps! As I progressed through the material, I constantly felt like studying was more difficult than it needed to be. I don't want shortcuts, I want to use a tool that better engages my memory without distractions.

ii. Information Overload and Inefficient Review Processes

Courses at my school give out an overwhelming amount of material for us to learn, often in a very short period of time. So much information that sometimes it's hard for me to know what to focus on. Reviewing for tests can be tedious and unstructured; I end up going over all the material including what I already know well instead of focusing on my weak spots. There are several tools that I've tried to use to help me learn. Quizlet is decent for flashcards but not

good at tracking my long term progress and spacing repetition. Anki is confusing to set up and almost overloaded with features that distract from the core activity of studying. Putting notes into Google Docs is freeform, I have forgotten where a lot of information I need to study goes. I can review the textbook for the class and come up with quizzes for myself, but I'm not really working on the weak points when the tools continually review very basic information. When I review, I want a system that helps me find and focus on my weak areas quickly without repetition on what I've already mastered.

iii. The Setup Time Problem: Friction Between Me and Learning

Trying to find what I need to study or spending too much time setting up and learning a system breaks the flow of my studying and takes up too much of the limited amount of time that I have to study in. I want something that can direct me to my problem areas quickly with a minimal amount of clicks between me and starting to study. Without this, it feels like I'm just reviewing topics that I already know. I waste time organizing notes, creating mini-quizzes for myself, and losing motivation due to poor feedback loops. I want to feel productive, and I don't want to feel like I spend more time organizing the information than actually doing the studying that I need to.

iv. Gamification Gone Wrong: When Features Become Obstacles

Most web apps try to gamify studying to hold engagement, but these distractions don't actually help me learn. Light gamification, such as streak tracking for consecutive days, is an easy way to encourage me to keep logging in, but heavier systems with leaderboards and achievements tends to add clicks and screens between me and what I want to accomplish. I want a system that I can get into, find a set of cards to study, and do the core activity: study. I want the system to allow me to do all this quickly and fit into a day where my class schedule and work schedule mean that I may not have a lot of time to sit at my device. A system with a clean

interface and useful, clearly signposted features will ensure that I don't spend too much time trying to get setup. It needs to work on any device that I have on me at the time, and not require that I sit at a computer to do the studying.

v. The Need for Intuitive Organization and Quick Setup

The system needs to be quick to set up. There should be an obvious way to create and edit cards, with controls to move them between decks. Additionally, setting up may take me too long. If I have a friend in the same class as me, I should be able to receive a deck shared by them, and bypass setting up a deck from scratch entirely. Although organization options would be nice, I should also be able to bypass those and start studying right away. I want a minimal amount of navigation between opening the web app and getting into a deck of flashcards to study. I should be able to go back and change tags and organization of decks as my needs change and I learn more about where I will need this information. There should be an option to move cards between decks and hide cards that I don't want to study. Organization in a study app should be intuitive with a minimum amount of clicks to get from logging into an account and getting to studying.

vi. Finding the Right Content: Search and Discovery Challenges

Conversely, finding flashcards on other apps can be a time consuming process in itself. There needs to be an easy way to find the deck that I currently want to study. These should be searchable by course, topic, or date created. This could possibly be implemented by some number of tags being allowed on each deck. Being able to customize and search on the name of the deck is important as well. It should be easy to review material, and the app should clearly report on scores so I can study the weak points of knowledge and start to work towards long term retention. It needs to be easy to customize. I want to be able to hide decks and cards I

don't want to see anymore, and I want to be able to move and duplicate cards between decks if needed.

vii. The Solution: A Tool That Actually Enhances Learning

If there was a better tool for studying, I could spend less time thinking about how to study, and more time actually learning the course material that I'm struggling with. It would reduce my stress before tests and exams by being more effective than the guesswork method or online tools that I currently use, increasing my retention and confidence. Long term, this would be a great method for me to review any subject that I've taken in school, or any certifications I want to study for after I have completed school. I don't need complicated settings or opaque connections between decks that take learning and time to set up. I need a study system that fits into my busy schedule and is intuitive enough to pick up immediately.

b. Decomposition into Sub-problems

Based on the customer narrative, the overall problem decomposes into:

1. **User Management:** Simple account creation and authentication system
2. **Content Creation:** Easy-to-use flashcard deck creation and editing
3. **Study Engine:** Spaced repetition algorithm with confidence-based scheduling
4. **Progress Tracking:** Basic analytics to show learning progress and maintain motivation
5. **Gamification:** Simple point system and streak tracking to encourage regular use
6. **User Interface:** Clean, mobile-responsive design optimized for studying

c. Glossary of Terms

| Term | Definition |
|----------------------|---|
| Deck | A collection of flashcards organized around a specific topic or subject |
| Flashcard | A digital card with a question on one side and answer on the other |
| Study Session | A period of active learning using flashcards with performance tracking |

| Term | Definition |
|--------------------------|---|
| Spaced Repetition | Learning technique where review intervals increase based on how well material is remembered |
| Confidence Rating | User's self-assessment (1-5 scale) of how well they remembered a flashcard answer |
| Active Recall | Learning method that requires retrieving information from memory rather than passive review |
| Study Streak | Number of consecutive days a user has completed study sessions |
| Mastery Level | Indicator of how well a user knows the material in a deck based on performance |
| Due Cards | Flashcards scheduled for review based on spaced repetition algorithm |
| Next Review | Calculated date/time when a flashcard should be shown again |

2. Goals, Requirements, and Analysis

a. Business Goals

Primary Goal: Create Simple, Effective Learning Tool

- Improve Student Learning Outcomes
 - Increase Information Retention (Target: 2x improvement over traditional methods)
 - Reduce Time Required for Effective Study
 - Make Studying More Engaging and Sustainable
- Maximize User Engagement

- Achieve High Daily Usage Rate\
- Maintain Simple, Intuitive User Experience
- Provide Motivational Features Without Complexity
- Demonstrate Technical Competency
 - Build Working MVP in 5 Weeks
 - Showcase Software Engineering Best Practices
 - Create Scalable Foundation for Future Development

b. Enumerated Functional Requirements

| ID | Priority | Description |
|-----------|-----------------|--|
| REQ-1 | High | User account creation and authentication |
| REQ-2 | High | Create and edit flashcard decks |
| REQ-3 | High | Add, edit, and delete individual flashcards |
| REQ-4 | High | Study flashcards with spaced repetition algorithm |
| REQ-5 | High | Rate confidence level for each flashcard (1-5 scale) |

| ID | Priority | Description |
|--------|----------|---|
| REQ-6 | Medium | Track study sessions and calculate progress |
| REQ-7 | Medium | Display user dashboard with deck overview |
| REQ-8 | Medium | Calculate and display study streaks |
| REQ-9 | Medium | Basic point system for completed study sessions |
| REQ-10 | Low | Simple leaderboard (if time permits) |

c. Enumerated Nonfunctional Requirements

| ID | Priority | Description |
|--------|----------|--|
| REQ-11 | High | Usability: New users can create first deck and study within 5 minutes |
| REQ-12 | Medium | Performance: Page load times under .3 seconds on mobile |
| REQ-13 | High | Reliability: System works consistently without crashes during demos |
| REQ-14 | High | Functionality: Support up to 100 concurrent users (adequate for class demo) |
| REQ-15 | Medium | Supportability: Clear error messages and user feedback |

| ID | Priority | Description |
|--------|----------|--|
| REQ-16 | Medium | Portability: Responsive design works on desktop and mobile browsers |
| REQ-17 | Low | Scalability: Code structure allows for future feature additions |

d. User Interface Requirements

| ID | Priority | Description | Visual Reference |
|------|----------|--|--|
| UI-1 | High | Clean, distraction-free study interface | Large flashcard display with minimal UI elements |
| UI-2 | High | Simple deck creation with intuitive form | Step-by-step process with clear field labels |
| UI-3 | High | Mobile-responsive design for all screens | Touch-friendly buttons, readable text on small screens |
| UI-4 | Medium | Dashboard showing deck status and statistics | Card-based layout with progress indicators |
| UI-5 | Medium | Visual feedback for confidence rating | 5-star rating system with immediate visual response |

3. Use Cases

a. Stakeholders

Development team:

The development team will be creating the project reports, algorithms, database tables, and developing the front end user interface. The team will also conduct program testing and user interface testing to ensure a reliable and user-friendly product that meets the customer requirements. The dev team's goal is to efficiently design and produce the required systems and produce a final product that meets the requirements of the other stakeholders. Additionally, the team must collaborate effectively, adhere to deadlines, and adapt to feedback during the design process.

Professor Vu:

Professor Vu serves as the project supervisor and mentor for the development team, making sure that project reports are of high quality and turned in on time. Responsibilities include reviewing and assessing the documentation, providing feedback on progress, and ensuring that deliverables are submitted on time and meet academic and professional standards. Professor Vu's goal is to guide the development team in the process of creating a quality, well thought out product.

End Users:

The end users are the stakeholders that represent the target audience for the software platform. Their main goal is to receive a product that functions correctly, is delivered on time, and improves their experience with studying.

b. Actors and Goals

| Actor | Type | Primary Goals |
|----------|---------------|---|
| User | Primary Actor | <ul style="list-style-type: none">-Interact with system-Edit and view cards-Receive timely, accurate responses from system |
| Browser | System Actor | <ul style="list-style-type: none">-Render interface accurately-Facilitate communication between user and server-Execute client-side scripts |
| Database | System Actor | <ul style="list-style-type: none">-Store and retrieve cards-Store and retrieve user information |

c. Use Cases

i. Casual Description

UC-1: Create Flashcard Deck (REQ-2, REQ-3)

User creates a new collection of flashcards on a specific topic by entering a deck name and adding question/answer pairs.

UC-2: Study Flashcards (REQ-4, REQ-5, REQ-6)

User reviews flashcards due for study, rates confidence level for each card, and system schedules future reviews based on spaced repetition algorithm.

UC-3: View Progress Dashboard (REQ-7, REQ-8, REQ-9)

User views their study statistics, including deck progress, study streaks, and points earned.

UC-4: Edit Flashcard Content (REQ-3)

User modifies existing flashcards by changing questions, answers, or deleting cards from their decks.

UC-5: Authenticate User (REQ-1)

User creates account and logs in.

UC-6: Rate Confidence (REQ-3)

User rates their confidence with studying individual flash cards to reduce visibility while viewing deck.

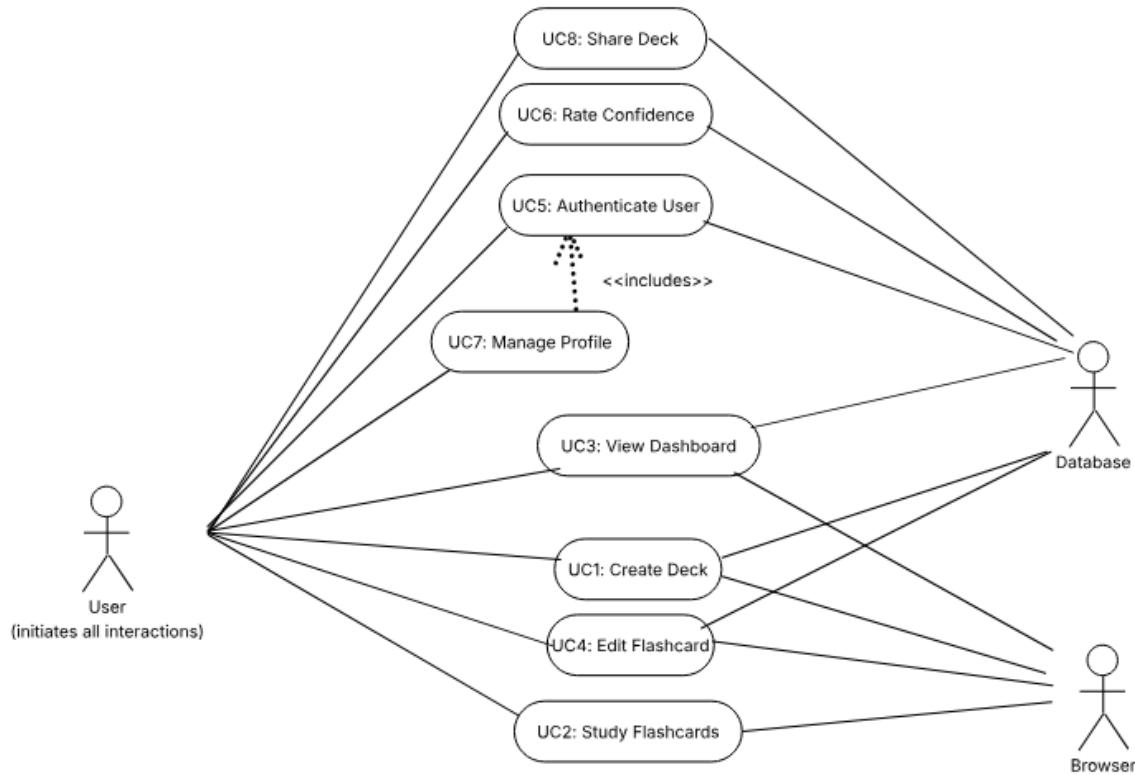
UC-7: Manage Profile (REQ-6, REQ-7)

User can see profile statistics and make changes to email and displayed name.

UC-8: Share Deck (REQ-10)

User shares a deck with another user via email link.

ii. Use Case Diagram



iii. Traceability Matrix

| Req't | PW | UC-1 Create Deck | UC-2 Study Cards | UC-3 View Dashb oard | UC-4 Edit Cards | UC-5 User Auth | UC-6 Rate Confid ence | UC-7 Manag e Profile | UC-8 Share Deck |
|---------------|----------|------------------------|------------------------|-------------------------------|-----------------------|----------------------|--------------------------------|-------------------------------|-----------------------|
| REQ-1 | 5 | X | | | | X | | | |
| REQ-2 | 5 | X | | | | | | | X |
| REQ-3 | 5 | X | | | X | | | | |
| REQ-4 | 5 | | X | | | | | | |
| REQ-5 | 4 | | X | | | | X | | |
| REQ-6 | 4 | | X | | | | X | X | |
| REQ-7 | 3 | | | X | | | | X | |
| REQ-8 | 2 | | | X | | | | | |
| REQ-9 | 2 | | | X | | | | | |
| REQ-10 | 1 | | | | | | | | X |
| UI-1 | 4 | | X | | | | | | |
| UI-2 | 4 | X | | | X | | | | |
| UI-3 | 3 | X | X | X | X | X | | | |
| UI-4 | 3 | | | X | | | | | |
| UI-5 | 2 | | X | | | | X | | |
| Max PW | 5 | 5 | 5 | 3 | 5 | 5 | 4 | 0 | 0 |

| Req't | PW | UC-1 Create Deck | UC-2 Study Cards | UC-3 View Dashb oard | UC-4 Edit Cards | UC-5 User Auth | UC-6 Rate Confid ence | UC-7 Manag e Profile | UC-8 Share Deck |
|----------|----|------------------------|------------------------|-------------------------------|-----------------------|----------------------|--------------------------------|-------------------------------|-----------------------|
| Total PW | 55 | 17 | 20 | 11 | 10 | 5 | 6 | 0 | 0 |

iv. Fully-Dressed Descriptions:

| | |
|--|--|
| Element Use Case UC-2: Study Flashcards | Description |
| Primary Actor | Student |
| Goal | Review flashcards using spaced repetition for optimal learning |
| Preconditions | User is logged in and has at least one deck with flashcards |
| Main Flow Steps | <ol style="list-style-type: none"> 1. User selects a deck from their dashboard 2. System calculates which cards are due for review based on spaced repetition algorithm 3. If cards are due, system displays first flashcard showing only the question 4. User reads question and attempts to recall answer mentally 5. User clicks "Show Answer" button 6. System reveals the correct answer 7. System presents confidence rating scale (1-5 stars) 8. User selects confidence level based on how well they remembered 9. System calculates next review date using spaced repetition algorithm 10. System updates study statistics (cards reviewed, session time) |

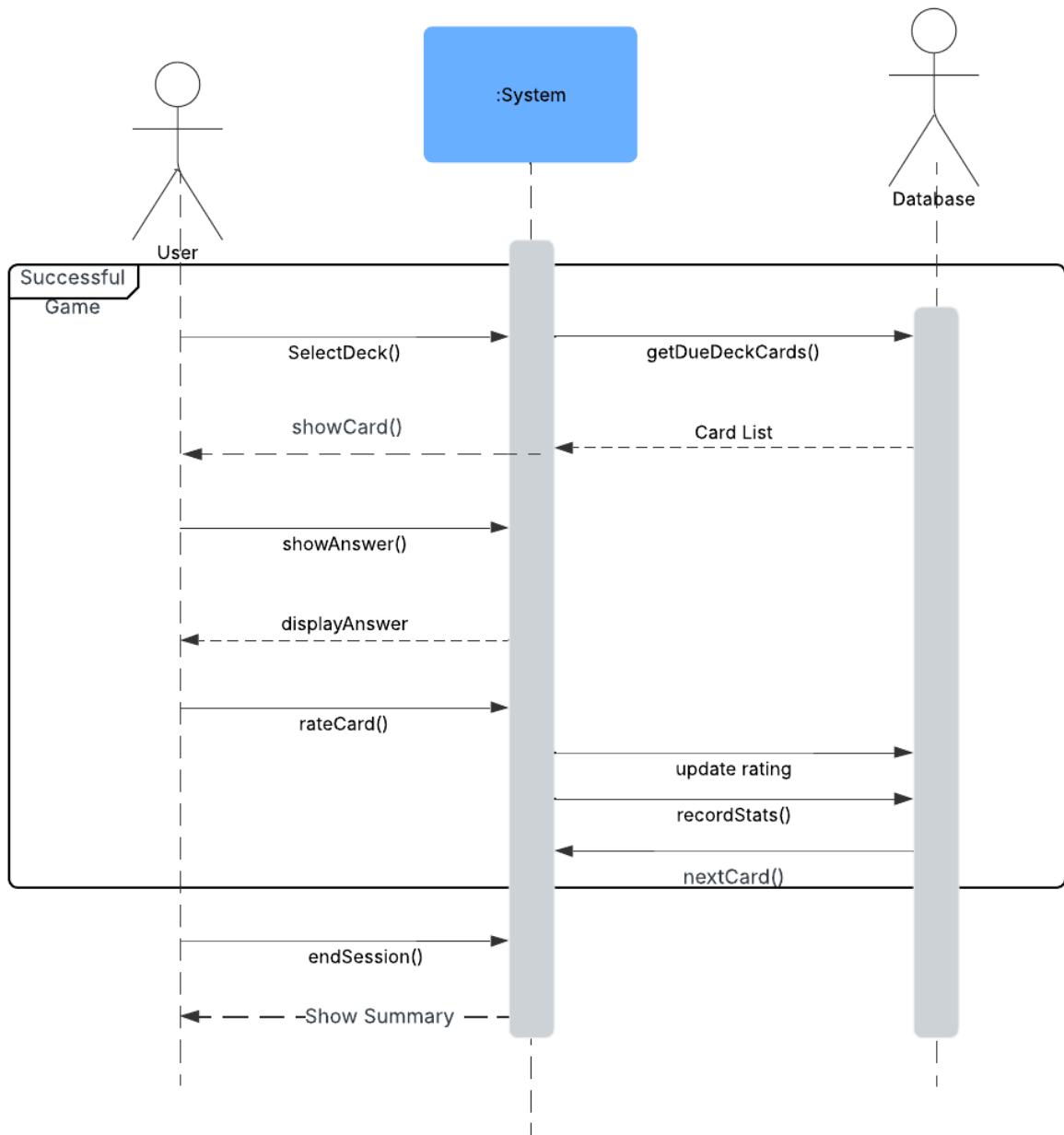
| Element Use Case UC-2: Study Flashcards | Description |
|--|--|
| | <p>11. System displays next due card or session completion if no more cards</p> <p>12. Repeat steps 3-11 until all due cards are reviewed</p> <p>13. System shows session summary with performance metrics</p> |
| Postconditions | Study session is recorded, flashcard schedules are updated, user statistics are incremented |

| Element Use Case UC-1: Create Flashcard Deck | Description |
|---|--|
| Primary Actor | Student |
| Goal | Create a new collection of flashcards for studying a specific topic |
| Preconditions | User is logged in |
| Main Flow Steps | <ol style="list-style-type: none"> 1. User clicks "New Deck" button from dashboard 2. System displays deck creation form 3. User enters deck name and optional description 4. User clicks "Create Deck" button 5. System creates deck and displays flashcard entry interface 6. User adds first flashcard by entering question and answer 7. User clicks "Add Card" button 8. System saves flashcard and displays empty form for next card |

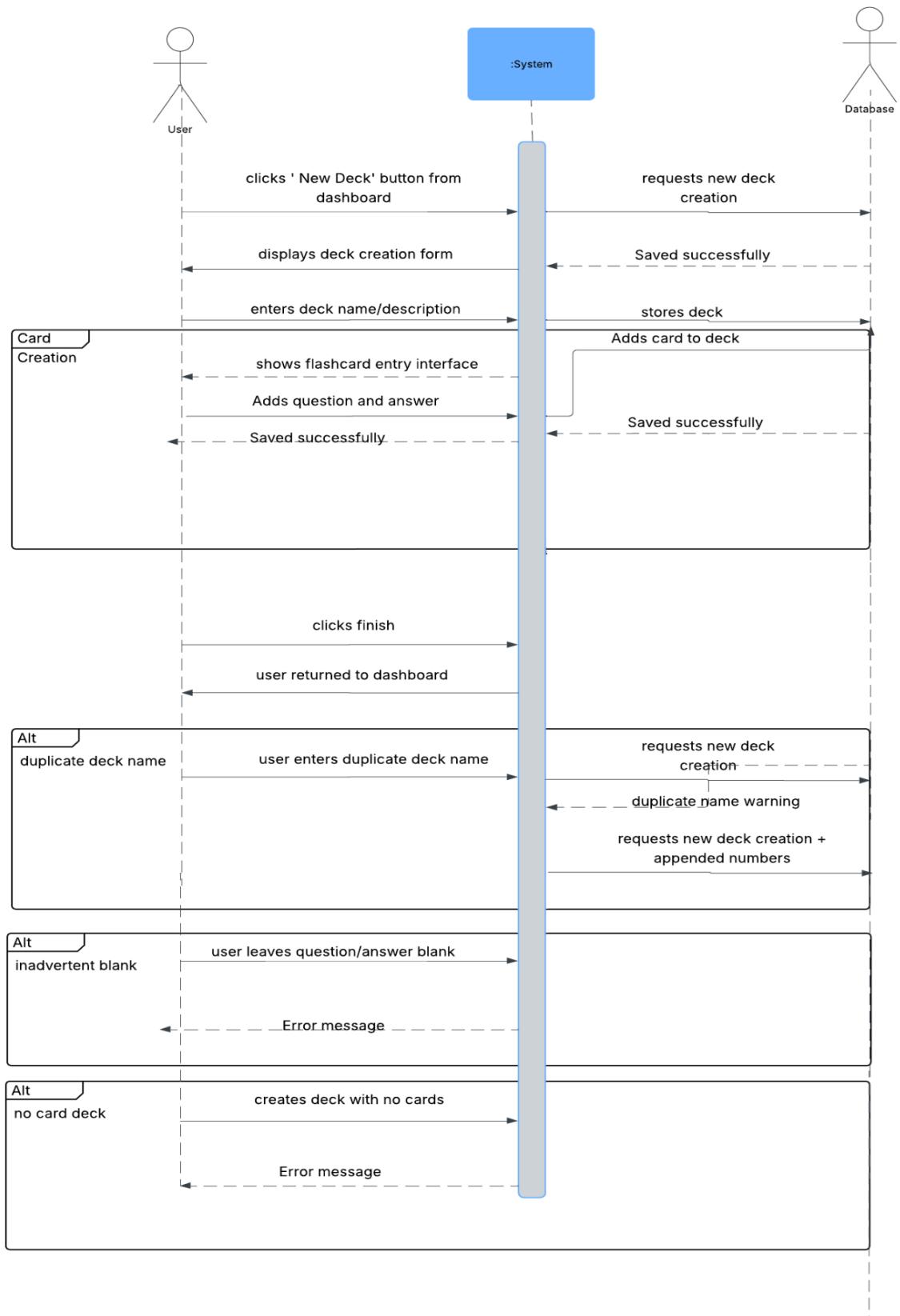
| Element Use Case UC-1: Create Flashcard Deck | Description |
|---|--|
| | <p>9. Repeat steps 6-8 until user has added desired number of cards</p> <p>10. User clicks "Finish" or "Start Studying" button</p> <p>11. System saves deck and returns user to dashboard or study interface</p> |
| Alternative Steps | <p>1. User enters duplicate deck name: System appends number to make unique</p> <p>2. User leaves question or answer blank: System displays error message</p> <p>3. User creates deck with no cards: System prompts to add at least one card</p> |
| Postconditions | New deck is created and available for study, user can add more cards or begin studying |

d. System Sequence Diagrams

UC-2: Study Flashcards



UC-1: Create Flashcard Deck

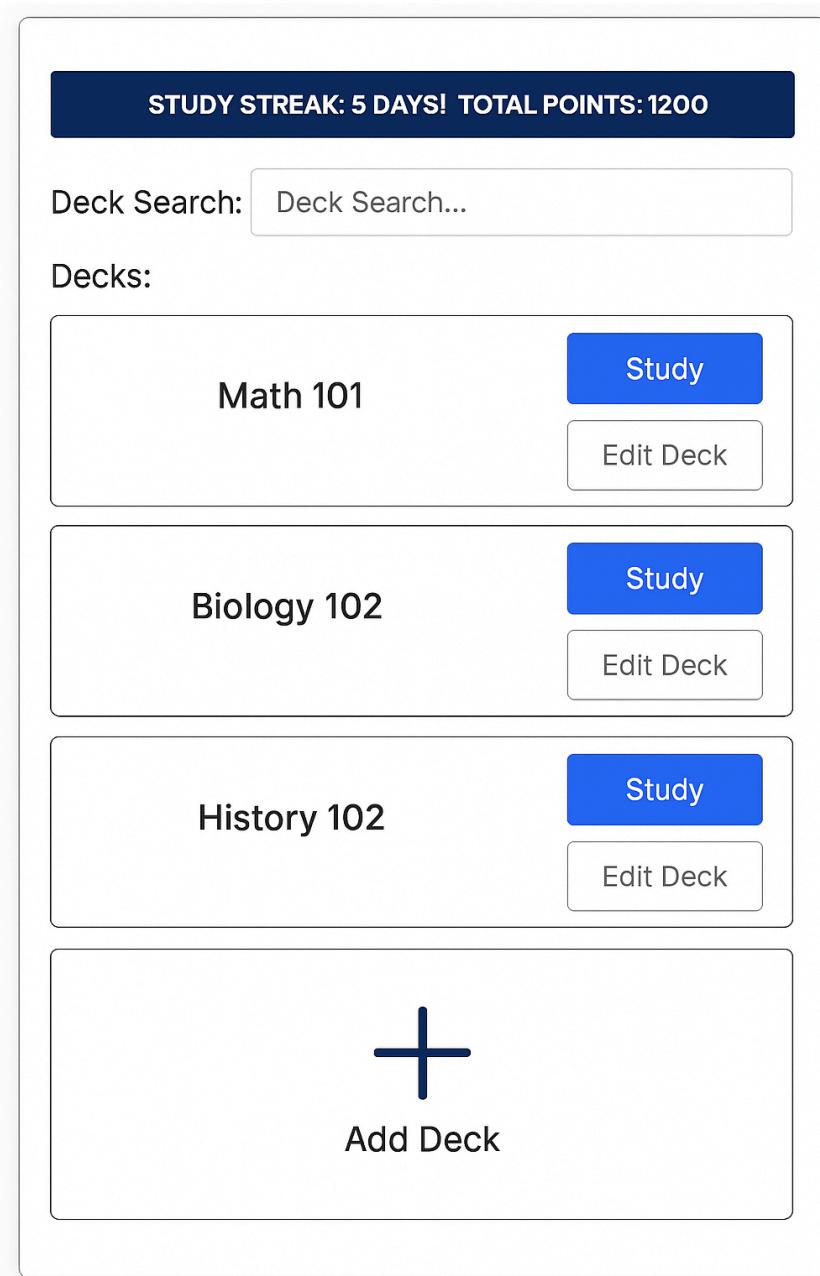


4. User Interface Specification

a. Preliminary Design

UC-1: Create Flashcard Deck

Step 1: Deck Creation



- The user clicks on "Add Deck"

Step 2: Add Flashcards

Quiz of Fury - Create New Deck

Deck Title:

Flashcards:

Question:

Answer:

Question:

Answer:

Add Flashcard

Save Deck

- The user enters information for deck title, questions and answers.
- The user saves the deck

Step 2 (alt for existing deck but new flashcards):

Quiz of Fury - Edit Deck

Biology 102

Question: What is the powerhouse of the cell?

Answer: Mitochondria

Delete Card

After clicking "Add Flashcard" a new Q/A box will pop up

Question:

Answer:

Add Flashcard

Save

UC-2: Study Flashcards

Step 1: Deck Selection

STUDY STREAK: 5 DAYS! TOTAL POINTS: 1200

Deck Search:

Decks:

Math 101

Study

Edit Deck

Biology 102

Study

Edit Deck

History 102

Study

Edit Deck

Add Deck



- The user clicks Biology to study Biology

Step 2 & 3: Study Interface - Question - Study Interface - Answer & Rating

Quiz of Fury - Study Session

Biology 101

What is the powerhouse of the cell?

Show Answer

..... after clicking show answer

Mitochondria

Confidence Rating:



Next Flashcard

Step 4: Session Complete

Quiz of Fury - Session Summary

Biology 101

You studied 15 cards!

Accuracy: 80%

Time Spent: 6 Minutes

Average Confidence: 4

[Home](#)

[Study Again](#)

b. User Effort Estimation

UC-1: Create Deck with 10 Cards

| Action | Clicks | Keystrokes | Time |
|------------------------|--------|------------|------|
| Navigate to "New Deck" | 1 | 0 | 5s |
| Enter deck name | 1 | 15 | 10s |
| Enter description | 1 | 30 | 15s |

| Action | Clicks | Keystrokes | Time |
|--|-----------|------------|---------------|
| Create deck | 1 | 0 | 2s |
| Add 10 cards (avg 20 chars each field) | 20 | 400 | 8 min |
| Finish deck | 1 | 0 | 2s |
| Total | 24 | 445 | ~9 min |

Breakdown: 5% navigation clicks, 95% data entry

UC-2: Study Session with 15 Cards

| Action | Clicks | Keystrokes | Time |
|-------------------------------------|-----------|------------|---------------|
| Select deck from dashboard | 1 | 0 | 5s |
| Study 15 cards (show answer + rate) | 30 | 0 | 6 min |
| Complete session | 1 | 0 | 5s |
| Total | 32 | 0 | ~6 min |

Breakdown: 100% navigation clicks, 0% data entry

UC-3: View Dashboard

| Action | Clicks | Keystrokes | Time |
|-------------------------|----------|------------|------------|
| Navigate to dashboard | 1 | 0 | 2s |
| Browse deck information | 0 | 0 | 30s |
| Total | 1 | 0 | 32s |

Breakdown: 100% navigation clicks, 0% data entry

5. Project Management

Project Timeline Overview

Project Duration: 5 weeks (June 12 - July 18, 2025)

Demo Dates: First Demo (July 8), Final Demo (July 18)

Sprint Planning

Sprint 1: Foundation Setup (June 12-19, 2025)

Goal: Establish working development environment and basic user authentication

Week 1 Deliverables:

- Django project structure with Docker setup
- User authentication system (login/logout/register)
- Basic database models defined
- Initial HTML templates and styling framework

Daniel Seip (Algorithm/Backend):

- Research and implement spaced repetition algorithm
- Design database schema for study tracking
- Implement user authentication system
- Create basic URL routing structure

Daniel Terreros (Backend/DevOps):

- Configure Docker development environment
- Set up Django project with proper settings
- Design database schema for study tracking
- Create core model relationships
- Set up database models and migrations

Julia Rogers (Frontend Lead):

- Design responsive CSS framework
- Create base HTML templates

- Implement authentication UI
- Set up static file handling

Sprint 1 Demo: Working login system with basic UI

Current Status: Functional Django app showing base and index templates for the quiz.

Next Steps: Git meeting to go over branching and commit rules to ensure a clean codebase

Sprint 2: Core Functionality (June 19-26, 2025)

Goal: Implement deck creation and basic study functionality

Week 2 Deliverables:

- Flashcard deck creation and editing
- Basic study interface (show question/answer)
- Simple confidence rating system
- Dashboard showing user's decks

Daniel Seip (Algorithm/Backend):

- Create study session tracking
- Build confidence rating processing
- Add basic statistics calculations

Daniel Terreros (Backend/DevOps):

- Implement deck creation views and forms
- Create flashcard CRUD operations
- Build basic study session logic
- Add deck management functionality

Julia Rogers (Frontend Lead):

- Create deck creation UI
- Design and implement study interface
- Build dashboard layout
- Add interactive elements and animations

Sprint 2 Demo: Can create decks and study with basic spaced repetition

Sprint 3: Polish and Gamification (June 26-July 3, 2025)

Goal: Add motivational features and improve user experience

Week 3 Deliverables:

- Point system and study streaks
- Improved dashboard with statistics
- Session completion feedback
- Mobile responsiveness testing

Daniel Seip (Algorithm/Backend):

- Fine-tune spaced repetition algorithm
- Add advanced statistics calculations
- Implement session management
- Create data backup and recovery

Daniel Terreros (Backend/DevOps):

- Implement points and streak tracking
- Create session statistics and reporting
- Add data validation and error handling
- Performance optimization

Julia Rogers (Frontend Lead):

- Design gamification UI elements
- Improve mobile responsiveness
- Add visual feedback and animations
- Conduct usability testing

Sprint 3 Demo: Engaging study experience with motivational features

Sprint 4: Testing and Demo Preparation (July 3-8, 2025)

Goal: Prepare for first demo with thorough testing

Week 4 Deliverables:

- Comprehensive testing (unit, integration, user)

- Bug fixes and stability improvements
- Demo preparation and presentation materials
- Documentation updates

All Team Members:

- **Testing:** Unit tests, integration tests, user acceptance testing
- **Bug Fixes:** Address issues found during testing
- **Demo Prep:** Create demo script, test scenarios, presentation
- **Documentation:** Update README, user guide, code comments

First Demo (July 8): Present working MVP to class

Sprint 5: Final Polish and Presentation (July 8-18, 2025)

Goal: Final refinements and presentation preparation

Week 5 Deliverables:

- Address feedback from first demo
- Final bug fixes and optimizations
- Complete documentation
- Final presentation preparation

Stretch Goals (if time permits):

- Simple leaderboard functionality
- Deck import/export feature
- Enhanced mobile app experience
- Social sharing capabilities
- Spaced repetition algorithm implementation
-

All Team Members:

- **Feedback Integration:** Implement suggestions from first demo
- **Final Polish:** UI improvements, performance tuning
- **Documentation:** Complete technical documentation
- **Presentation:** Prepare final demo and presentation materials

Final Demo (July 18): Present completed project

Team Responsibilities and Coordination

Role Definitions

Daniel Seip - Algorithm/Backend & Project Manager:

- Study session logic and data processing
- Performance optimization and monitoring
- Project coordination and timeline management
- Code review and quality assurance

Daniel Terreros - Algorithm Specialist & DevOps:

-
- Overall project architecture and Django setup
- User authentication and account management
- Database design and model relationships
- Deployment and environment management
- Technical problem-solving and debugging

Julia Rogers - Frontend Lead & UX Designer:

- User interface design and implementation
- Responsive web development
- User experience optimization
- Usability testing and feedback integration
- Visual design and branding

Communication and Coordination

Daily Standups: 15-minute check-ins via Discord/Slack (Monday, Wednesday, Friday) **Sprint Reviews:** End-of-week demos and retrospectives **Code Reviews:** All commits reviewed by at least one other team member **Documentation:** Shared Google Drive for requirements, design decisions, and meeting notes

Risk Mitigation

Technical Risks:

- **Docker setup issues:** Pair programming sessions for environment setup
- **Algorithm complexity:** Start with simple implementation, iterate
- **Integration problems:** Daily integration testing

Timeline Risks:

- **Scope creep:** Strict adherence to MVP feature list
- **Individual delays:** Cross-training on critical components
- **External dependencies:** Buffer time built into each sprint

Conclusion and Project Evaluation

The Quiz of Fury project provides a focused, user-friendly platform designed to help students study more efficiently using spaced repetition and active recall. Its clean design and confidence based tracking support meaningful engagement without overwhelming the user. The application's simple structure makes it ideal for continued development and potential adoption in educational settings.

Pros of the project include its intuitive interface, mobile responsiveness, and motivational features like study streaks and point tracking. It also offers a solid technical foundation built with Django, allowing for scalability and long term maintenance. Cons include limited initial features like social sharing or advanced analytics, which could be added in future iterations.

The estimated development cost for a basic MVP is minimal, as the project was completed by a small development team within five weeks using open source tools. If expanded commercially, projected costs would primarily involve hosting, maintenance, and possible marketing. The project could generate revenue through premium features, educational licensing, or ad-supported free versions, allowing the company to recover costs and grow over time.

With its clear value and potential, the project is well positioned for further investment. Without a positive conclusion projects often risk being abandoned, but Quiz of Fury demonstrates both educational impact and long-term promise.

References

- Django Software Foundation. (2025). *Django Documentation*. Retrieved from <https://docs.djangoproject.com/>
- Quizlet. (2025). *Quizlet Help Center*. Retrieved from <https://help.quizlet.com/>

- Anki. (2025). *Anki Manual*. Retrieved from <https://docs.ankiweb.net/>

Revisions

- User interface mockups redone
- Some requirement priorities modified
- Title page updated for a more professional presentation
- Added conclusion
- Modified project management timeframe
- Revised the use case to requirement matrix