

Names

Tuesday, 29 April 2025 12:01 PM

Danielle Dormer - n11461047

Emily Leonard - n11959983

Leila Raad - n11457333

Olivia Miley - n11558555

Max Rasmussen - n1201348

To Do Page

Monday, 24 March 2025 12:10 PM

(Transition this into GitHub projects)

To Do	In progress	Done
	<p>CHECKPOINT 2 - User Stories (20 total, 4 each)</p> <ul style="list-style-type: none">• OM - High-school students (Competitive, easily overwhelmed, gets bored easily, no time)• LR - Uni students (No time, forgetful, appreciates aesthetics, overwhelmed)• EL - High-school students (Forgetful, visually impaired, time management, appreciates aesthetics)• MR - Uni students (Competitive, bored easily, visually impaired, time management)• DD - Teachers	CHECKPOINT 2 - Project Management Tool (GitHub Projects)
	<p>CHECKPOINT 2 - UI Design (Medium Fidelity)</p> <ul style="list-style-type: none">• DD - medium-fidelity• MR - medium/high-fidelity	<p>CHECKPOINT 2 - UI Design (Low Fidelity)</p> <ul style="list-style-type: none">• OM - low-fidelity• LR - low-fidelity• EL - medium-fidelity
		<p>CHECKPOINT 2 - Release Plan and First Sprint Plan</p> <ul style="list-style-type: none">• Everyone during group meeting

Assessment Requirements

Monday, March 24, 2025 11:59 AM

Week 9: Preliminary Prototype minimal requirements:

Project Requirements:

- *Preliminary (functional) Prototype: The project has a functional user interface and stores data to a DB for persistence*
- *Test-Driven Development: the project includes a suitable test suite*
- *Version Control Workflow: There is evidence of regular commits to repositories and proper source management practices (e.g. branching)*
- *Object Oriented Design: Initial use of Object Oriented design patterns(e.g. MVC) and principles (e.g. Encapsulation, Inheritance, ...)*

Walkthrough video contents:

- explain the benefit of the project
- overview of user stories and prioritisation
- low fidelity prototypes of the User Interface
- overview of Github and PM tool, including discussion of assigned tasks
- Brief overview of the Java code created so far with emphasis on
 - JavaFX user interface
 - Persistence (DB)
 - Unit Tests
- Additionally, for excellent marks
 - include a 1 minute demo showing the project in action. Demonstrate use of the functionalities implemented (preliminary GUI and DB)

Week 13: Final Prototype and Demo minimal requirements

Project Requirements:

- *The Final Prototype delivers the minimal requirements:*
 - a graphical user interface with windows for the main functionalities, in general based on JavaFX
 - an authentication system to sign-up/sign-in (GUI and models)
 - a persistency system to store/retrieve/update user data (GUI and models)
 - one or (more likely) more application windows in which the actual, useful part of the application is performed (again, with GUI and models).
- *All 'Must-Have' features have now been implemented as they were detailed in the user stories*
- *Build Script: Build and Tests are automated using a build script in Github*
- *Automated Build Server: Github 'Actions' are in place to automated the CI/CD process*
- *Documentation: Javadoc comments and generated pages.*
- *Object Oriented Design: Advanced use of Object Oriented design patterns (e.g. Dependency Injection, Builder, Singleton, Factory, Observer) and principles (abstraction, encapsulation, inheritance, and polymorphism)*

Walkthrough video contents:

- Detailed presentation of the Java code created so far with emphasis on
 - JavaFX user interface
 - Persistence (DB)

- Unit Tests
- Detailed presentation of the CI/CD and build automation scripts in Github
 - Github actions
 - Automated Tests/Builds
- Demo of the application in action
- Additionally, for excellent marks
 - Outcomes of the User Testing

Our Requirements

Monday, March 24, 2025 11:59 AM

1. The user can create multiple 'quests'.
2. The user can specify an end date for each quest.
3. The user can use the app to generate flashcards from studied content.
4. The user can add, edit, delete and review flashcards.
5. The user can complete daily quizzes based on the generated flashcards.
6. The user can progress in the 'quest' based on completion of daily quizzes.
7. The user can see high-scores from past 'quests'.

Project Brief

Monday, March 24, 2025 12:00 PM

Project Brief:

Our application (AlfredAI) will assist students with studying efficiently by gamifying the process. It will aim to increase productivity by displaying engaging visuals and incentives to continue study.

This app is a gamified study tool where, for each unit, you can start a new “quest.” The app generates flashcards for you and uses them to build daily quizzes. As you complete quizzes, you progress further in the quest, with your progress influenced by quiz scores and login streaks that give you bonus travel distance. Each quest has a set length (marked by a “final boss” deadline) and your progress, or distance travelled, is determined by how much studying you do each day. The more distance you cover, the more rewards and preparation you gain for the final boss. The app tracks high scores for the longest quests and fastest progress. Outside of quizzes, you can generate new flashcards from content you paste in, edit, delete and review existing flashcards.

Additional requirements we could add:

8. The user can earn rewards (such as outfits or items) as they progress in quests.
9. Every week the user can face a "mini-boss" which involves completing a quiz with all flashcards.

Link to repository: <https://github.com/eleon236/CAB302-AlfredAI-Project.git>

About the app:

- For each unit you can start a new “quest”
- The app can make flashcards for you and then use the flashcards to create daily quizzes for you
- As you complete daily quizzes you progress further in the quests (depending on your score in the quizzes) - streaks also add bonuses to how far forward you progress
- Each quest has a:
 - Length (determined by an end date you can set, when the ‘final boss arrives’ or something)
 - Distance travelled (determined by how much study you do each day - number of flashcards - streaks from logging in days in a row mean you can travel further)
- The further you travel the more cool stuff you get, so the more prepped you’ll be for the final boss!
- So then it records your high scores for furthest quest and furthest travelled in a short time
- The work you have to clock in will be:
 - Versing bad guys in quizzes based on the flashcards
 - The daily quiz will use a number of flashcards equal to the total flashcards divided by the remaining days in the quest
 - More likely to include flashcards that haven’t been mastered yet, but will
- Outside of quizzes you have to clock into daily, you can:
 - Generate more flashcards from copy-pasted content
 - Edit the flashcards - colour code? Sort into categories? Do either by hand or by asking the AI to do it for you (e.g. remove these ones, change this to be worded better, etc)
 - Review the flashcards

Release Plan

Monday, 7 April 2025 10:43 AM

The scope of the release will involve the entire project. All user stories prioritised as "must-have" or "should-have" should be addressed in this release.

The release date is 27 May, 2025.

The resources for this project are capped at 5 team members, and approximately 13 weeks with each team member contributing 5-10 hours of work per week.

Dependencies include Amazon corretto, AI API, Java fx, GitHub, the applications database, VPN

Potential risks include, the VPM and the AI might not work. In case they don't work we will have some base flashcards to run.

Notes About Writing the Release Plan:

- **Release Plan:** Not heavily emphasized in CAB302 since there's only one release (Week 13).

A release plan should cover five key areas:

1. **Scope:** Describes the features, enhancements, and bug fixes in the release.
2. **Timeline:** Specifies the intended release date.
3. **Resources:** Identifies the team members, tools, and infrastructure needed.
4. **Dependencies:** Addresses any external factors or prerequisites.
5. **Risk Assessment:** Evaluates potential risks and identifies mitigation strategies.

- Tutor couldn't really think of any risks we'd have to consider, other than the fact that we will run into more and more time pressure towards the end of the semester, so should have some kind of backup plans when things start to feel a bit overwhelming.
- Note down what could go wrong and what's your backup plan. E.g. having randomly generated responses from 50 pre-written ones, rather than implementing AI if it doesn't work well?
- E.g. something to add to risk assessment: will need to install a VPN to use the AI API - could possibly have issues with that.
- **OVERALL:** Release plan file will probably be pretty empty (upload to Github), except for all the user stories listed. Will just have one release, strict timeline (wk 13 submission), capped resources (team size, hours worked, etc.).

User Stories (All Combined)

Tuesday, March 25, 2025 11:43 AM

Template for user stories:

[https://canvas.gut.edu.au/courses/20355/pages/activity-5-dot-4-homework-create-user-stories?
module_item_id=1807969](https://canvas.gut.edu.au/courses/20355/pages/activity-5-dot-4-homework-create-user-stories?module_item_id=1807969)

Splitting up the user stories:

- Students (high-school & uni)
- Teachers

1. No time
2. Forgetful (need the end date)
3. Competitive (want high-scores)
4. Visually impaired (glasses, colour blind etc.)
5. Appreciates aesthetics
6. Gets bored easily - needs incentives and interactive way to learn
7. Gets overwhelmed by lots of content - struggles to make flashcards, would appreciate it summarised
8. Time management - leaves all study until the night before the exam (needs the incentive to study a bit each day)

Requirements:

1. The user can create multiple 'quests'.
2. The user can specify an end date for each quest.
3. The user can use the app to generate flashcards from studied content.
4. The user can add, edit, delete and review flashcards.
5. The user can complete daily quizzes based on the generated flashcards.
6. The user can progress in the 'quest' based on completion of daily quizzes.
7. The user can see high-scores from past 'quests'.
8. a graphical user interface with windows for the main functionalities, in general based on JavaFX
9. an authentication system to sign-up/sign-in (GUI and models)
10. a persistency system to store user data (GUI and models)
11. a persistency system to retrieve user data (GUI and models)
12. a persistency system to update user data (GUI and models)
13. one or (more likely) more application windows in which the actual, useful part of the application is performed (again, with GUI and models).
 1. Login
 2. Home
 3. Quests page
 4. Flashcards
 5. Exam page

Requirement	User Stories	Acceptance Criteria
1. The user can create multiple 'quests'. MH	As a student, I want to be able to separate each of my subjects' flashcards and unit material, so that I can study all my assigned subjects efficiently.	Given that there is an add subject button, when I click on this button, then I can fill out a form for my subjects.
SH	As a student, I want to select a different character to represent each of my subjects, so I can more easily differentiate between each unit's content.	Given different 'quests' can be created for each of the user's subjects, when I create a new quest, then I will be able to select a character for that quest.
2. The user can specify an end date for each quest. MH	As a high school student, I want an application that helps me manage my time by setting end dates for my Quests so that I can plan out my study time effectively before my exams.	Given that I need help managing my time, when I start studying I should set date for deadlines helping me stay organised and motivated to complete my studies on schedule.
MH	As a forgetful university student, I want to set an end date for each of my 'quests' which reflects my courses' final exam dates, so I won't forget when each of my final exams are.	Given multiple 'quests' can be created, when I create one of these quests, then I will be able to set an end date for it.
3. The user can use the app to generate flashcards from studied content. MH	As a busy university student with little free time, I want to generate flashcards quickly and easily, so that I can spend less time preparing to study.	Given there is an input in the application, when I paste my course content into the input, then the application will generate flashcards for me.
SH	As a Teacher. I want a feature to generate flashcards effortless based on my class topics. So I can spend less time making and more time teaching.	Given there is an place in the application to put in external resources into the application. That then will generate flashcards for the user.
MH	As a high school student that struggles with learning, I often get easily overwhelmed with the quantity and complexity of my subject contents. I want a feature that creates study material from notes. As, the process of creating notes is generally what overwhelms me. With this feature it will allow me to study easily and limit how	

Given that there is a catalogue tag on the home page & correct information
When I click on the tag
Then I can view product information & price

MoSCoW Prioritisation Framework

Used to categorise user stories based on importance:

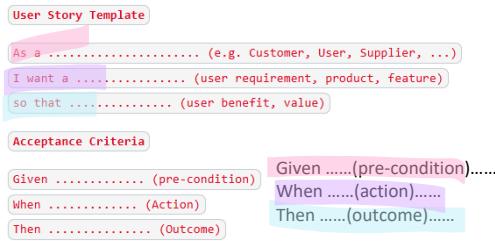
- **Must-have:** Critical to system function; project fails without it.
- **Should-have:** Important, but not vital for first delivery.
- **Could-have:** Nice to have; add if time/resources allow.
- **Won't-have:** Not planned for this release; maybe future work.

	material from notes: As, the process of creating notes is generally what overwhelms me. With this feature it will allow me to study easily and limit how overwhelming studying is for me.	
MH	As a forgetful high school student, I want an application that helps me review my flashcards daily so that I can stay on track with my learning and be better prepared for exams.	Given that I am forgetful, when I am going over my notes I can quickly add and jot down any notes so that later when I am completing my routine daily quizzes, Then I will be able to recall my notes
4. The user can add, edit, delete and review flashcards. CH	Given that I am visually impaired, it is a requirement of applications that I use to be able to use the windows narrator to know what the application is saying, as such I want a product	Given that I am visually impaired, when I use an application I expect accessibility stands so then I can use the product comfortably
MH	As a forgetful university student, I want to add more flashcards to quests over time, so I can add content I previously forgot about.	Given there is an input in the application and flashcards are already generated, when I paste more course content into the input , then the application will generate additional flashcards for my quest.
MH	As a university student, I want to edit generated flashcards, so I can make sure they are right for me after being generated.	Given flashcards are already generated, when I select a flashcard, then I will be able to edit it.
MH	As a university student, I want to delete generated flashcards, so I can make sure they correctly reflect course content if certain topics are irrelevant.	Given flashcards are already generated, when I select a flashcard, then I will be able to delete it.
MH	As an easily overwhelmed university student, I want to view flashcards which summarise course content, so I can understand the content without being overwhelmed.	Given flashcards can be generated, when I generate these flashcards, then I will be able to review these flashcards at any time.
MH	As a Teacher. I want to review the flashcards. So I can make sure that the flashcards align with my teaching style.	Given there is page in the application that gives an overview of the flashcards before they get published and used by the users.
MH	As a Teacher. I want to edit the flashcards. So I can change the flashcards so they aren't missing key words or subtopics.	Given there is a button called edit. When I press this button, I am allowed to edit the flashcards.
5. The user can complete daily quizzes based on the generated flashcards. SH	As a student that generally struggles with time management having a program that will allow me to time myself will then help me with managing my time.	Given that I struggle with time management when a program gives a timer to do actives and progress bar then I will be able to focus better.
MH	As a user of this application, I have the requirement that flashcards are generated from my subject notes as I often have no time to create these cards myself. Which often means I don't study effectively or effectively for my exams. So this requirement will ensure I am study all my material effectively despite my time constraints.	
CH	As a competitive student, I want to earn rewards such as in-game items for completing daily quizzes, so I can have more of an incentive to keep studying.	Given users can complete daily quizzes, when I complete each daily quiz, then I can receive an in-app reward.
MH	As a Teacher. I want to do a quiz based on the flashcards used for that specific class. So I can get everyone involved and guessing at the questions.	Given there is a button that is called quiz generation and once pressed generates a quiz based on the flashcards that are selected.
6. The user can progress in the 'quest' based on completion of daily quizzes. MH	As a university student who often gets bored while studying and is unable to keep at a task I want to be engaged throughout the learning process, so that I remain interested in my learning without getting bored or repetitive.	Given that I get bored studying and not able to engage, when a product that gamifies the experience then I can stay engaged in the studying,
SH	As a student I often get bored when having to study for an extended period of time to then later either give up or procrastinate till last minute. I want an application that helps me stay engaged in my learning so I can study before an exam without having to either take lots of breaks or not study at all, to achieve better grades for my exams.	
MH	As a competitive student, I want to see my progress on a completion bar, so I	Given users can progress in 'quests' based on completion of daily quizzes,

	can study more due to the incentive to progress further.	when I progress in my quests, then I will be able to see my progression on a progress bar.
7. The user can see high-scores from past 'quests'. CH	As a highly competitive student. I want a feature that allows me to compare me to compete either against myself or others. This is so that I am more motivated to learn and have a greater chance at engaging with my high school curriculum.	Given the user condition of wanting to compete against themselves as a form of engagement. When the user completes a 'quest' they are able to rechallenge that 'quest' in order to achieve a higher score, as all previous high scores are kept on record. Therefore, the user would be more inclined to keep studying in order to receive the highest score possible.
SH	As a high performing student I want to see my high scores and achievements so that I can gauge my performance and always see an improvement.	
WH	As a Teacher. I want to see high scores from past quizzes. So I can see what students did better at and what they didn't do well at.	Given there is a page that the user can see an overview of the weekly quiz scores to select to work on more.
8. A graphical user interface with windows for the main functionalities, in general based on JavaFX MH	As a student, I want to study using an app with an interface, so I can study more easily.	Given there is an app, when I open the app, then it will display a graphical user interface for me to interact with.
9. An authentication system to sign-up/sign-in (GUI and models) MH	As a student, I want to sign up for an account in the app, so my information can be saved to an account.	Given there is a sign-up button, when I click this button, then I can enter my details to create my account.
MH	As a student, I want to sign into the app, so I can see information saved to my account.	Given there is a sign-in page, when I enter my account details, then I can use the app signed into my account.
10. A persistency system to store user data (GUI and models) MH	As a student, I want to store my data, so I don't lose my account after closing and reopening the app.	Given users can create accounts in the app, when I make my account, then the app will store my data.
11. a persistency system to retrieve user data (GUI and models) MH	As a student, I want to retrieve my data, so that I can see my past flashcards and progress.	Given users can create accounts in the app, when I log into my account, then the app will retrieve all my account data.
12. a persistency system to update user data (GUI and models) MH	As a student, I want to update my stored data, so that I can update my past flashcards and progress.	Given users can create accounts in the app, when I make updates in my account such as generating new flashcards, then the app will store them.
13. one or (more likely) more application windows in which the actual, useful part of the application is performed (again, with GUI and models). MH	As a student, I want to use a study app that has multiple different windows, so my study platform can be organised.	Given users can study on the application, when I use the app, then I will see multiple different windows implemented.
SH	As a Student, I want to have a modern design with attractive visuals so that I can enjoy using the app while studying and feel motivated to continue learning.	Given that the design is well thought out, when I interact with the application then I should experience a clean, visually striking interface.
CH	As a visually impaired student, I want accessibility features like high contrast and screen reader support so that I can easily navigate through the app.	Given that the application is accessible for visually impaired individuals, when I use features, then the app should provide visual aids

Olivia Stories

Tuesday, 25 March 2025 11:58 AM



- Requirements:
1. The user can create multiple 'quests'.
 2. The user can specify an end date for each quest.
 3. The user can use the app to generate flashcards from studied content.
 4. The user can add, edit, delete and review flashcards.
 5. The user can complete daily quizzes based on the generated flashcards.
 6. The user can progress in the 'quest' based on completion of daily quizzes. (game mentality enticement)
 7. The user can see high-scores from past 'quests'.

OM - High-school students (Competitive, easily overwhelmed, gets bored easily, no time)

User Story	Acceptance Criteria
As a highly competitive student. I want a feature that allows me to compare me to compete either against myself or others. This is so that I am more motivated to learn and have a greater chance at engaging with my high school curriculum.	<i>The user can see high-scores from past 'quests'.</i> Given the user condition of wanting to compete against themselves as a form of engagement. When the user completes a 'quest' they are able to rechallenge that 'quest' in order to achieve a higher score, as all previous high scores are kept on record. Therefore, the user would be more inclined to keep studying in order to receive the highest score possible.
As a high school student that struggles with learning, I often get easily overwhelmed with the quantity and complexity of my subject contents. I want a feature that creates study material from notes. As, the process of creating notes is generally what overwhelms me. With this feature it will allow me to study easily and limit how overwhelming studying is for me.	<i>The user can use the app to generate flashcards from studied content.</i>
As a student I often get board when having to study for an extended period of time to then later either give up or procrastinate till last minute. I want an application that helps me stay engaged in my learning so I can study before an exam without having to either take lots of breaks or not study at all, to achieve better grades for my exams.	<i>The user can progress in the 'quest' based on completion of daily quizzes.</i>
As a user of this application, I have the requirement that flashcards are generated from my subject notes as I often have no time to create these cards myself. Which often means I don't study effectively or effectively for my exams. So this requirement will ensure I am study all my material effectively despite my time constraints.	<i>The user can complete daily quizzes based on the generated flashcards.</i>

Completing
the exam

1. The user can create multiple 'quests'.	As a student, I want to be able to separate each of my subjects' flashcards and unit material, so that I can study all my assigned subjects efficiently.	Given that there is an add subject button. When I click on this button. Then I can fill out a form for my subjects.
8. A graphical user interface with windows for the main functionalities, in general based on JavaFX		Given, --- then I can easily interact with it.
9. An authentication system to sign-up/sign-in (GUI and models)		
10. A persistency system to store user data (GUI and models)		
11. A persistency system to retrieve user data (GUI and models)		
12. A persistency system to update user data (GUI and models)		
13. One or (more likely) more application windows in which the actual, useful part of the application is performed (again, with GUI and models).		

Many multiple windows for different functions

Sign up
if username in database THEN
render user
else render none

length
Search username get row of info
if search = 0 return
incorrect username Please try again

```
if userame in database  
    render userame  
else check password  
    if passwords don't match  
        Passwords don't match reenter  
    else  
        upload to database.
```

if search = 0 returns incorrect username Place to my session

else
 if entered password = database
 Continue to application
 Store user ID to global variable
 to be used later for easier access
 else
 Password entered is not password

add user

get user ID

add quest

update quest distance

quest last quiz date

last quest

add flashcard

update flashcard

delete plusnews

quest followed

Not the guest

Danielle Stories

Tuesday, 25 March 2025 12:03 PM

User Story Template

As a (e.g. Customer, User, Supplier, ...)

I want a (user requirement, product, feature)

so that (user benefit, value)

Acceptance Criteria

Given (pre-condition)

When (Action)

Then (Outcome)

Teachers

1. Want to teach the class with flashcards
2. Visually impaired (glasses, colour blind etc.)
3. Appreciates aesthetics
4. Time to recap the class learnt with a quiz based on the flashcards

Requirements	User Story	Acceptance Criteria
The user can use the app to generate flashcards from studied content.	As a Teacher. I want a feature to generate flashcards effortless based on my class topics. So I can spend less time making and more time teaching.	Given there is an place in the application to put in external resources into the application. That then will generate flashcards for the user.
The user can add, edit, delete and review flashcards.	As a Teacher. I want to review the flashcards. So I can make sure that the flashcards align with my teaching style. As a Teacher. I want to edit the flashcards. So I can change the flashcards so they aren't missing key words or subtopics.	Given there is page in the application that gives an overview of the flashcards before they get published and used by the users. Given there is a button called edit. When I press this button, I am allowed to edit the flashcards.
The user can complete daily quizzes based on the generated flashcards.	As a Teacher. I want to do a quiz based on the flashcards used for that specific class. So I can get everyone involved and guessing at the questions.	Given there is a button that is called quiz generation and once pressed generates a quiz based on the flashcards that are selected.
The user can see high-scores from past 'quests'.	As a Teacher. I want to see high scores from past quizzes. So I can see what students did better at and what they didn't do well at.	Given there is an page that the user can see an overview of the weekly quiz scores to select to work on more.

Max Stoires

Tuesday, 25 March 2025 12:05 PM

University Students

1. As a high performing student I want to see my high scores and achievements so that I can gauge my performance and always see an improvement.
2. As a university student who often gets bored while studying and is unable to keep at a task I want to be engaged throughout the learning process, so that I remain interested in my learning without getting bored or repetitive.
3. Given that I am visually impaired, it is a requirement of applications that I use to be able to use the windows narrator to know what the application is saying, as such I want a product
 - *Change to be high contrast and large font that is able to en-corporate this to benefit my learning.*
4. As a student that generally struggles with time management having a program that will allow me to time myself will then help me with managing my time.

Emily Stories

Tuesday, 25 March 2025 12:05 PM

(Forgetful, visually impaired, time management, appreciates aesthetics)

User Story	Acceptance Criteria	Requirement
As a high school student, I want an application that helps me manage my time so that I can plan out my study time effectively before my exams.	Given that I need help managing my time, when I start studying I should set date reminders for deadlines helping me stay organised and motivated to complete my studies on schedule.	2 and 6
As a forgetful high school student, I want an application that helps me review my flashcards daily so that I can stay on track with my learning and be better prepared for exams.	Given that I am forgetful, when I am going over my notes I can quickly add and jot down any notes so that later when I am completing my routinely daily quizzes, Then I will be able to recall my notes	4 and 5
As a high school student who enjoys a visually appealing and engaging study experience, I want the app to have a modern design with attractive visuals so that I can enjoy using the app while studying and feel motivated to continue learning.	Given that I appreciate aesthetically pleasing designs, when I interact with the app, then I should experience a clean, visually striking interface with animations that make studying feel more enjoyable and engaging, keeping me interested in the process.	*Aesthetics requirement
As a visually impaired high school student, I want the app to have accessibility features like high contrast mode and screen reader support so that I can easily navigate through the app, read my flashcards, and complete quizzes without difficulty.	Given that I have a visual impairment, when I use the app, then the app should provide a high contrast interface for flashcards and quizzes, and all interactive elements should be screen reader compatible, ensuring I can study independently.	*Usability requirement
As a high school student, I need an application that allows me to study for multiple subjects, so that I am able to study for all my assigned subjects	Given that I have many different subjects, when I am studying, I need a section for each of my enrolled subjects. Then I will be able to study effectively.	1

User Story Template

As a (e.g. Customer, User, Supplier, ...) I want a (user requirement, product, feature) so that (user benefit, value)

Acceptance Criteria

Given (pre-condition)
When (Action)
Then (Outcome)

1. The user can create multiple 'quests'.
2. The user can specify an end date for each quest.
3. The user can use the app to generate flashcards from studied content.
4. The user can add, edit, delete and review flashcards.
5. The user can complete daily quizzes based on the generated flashcards.
6. The user can progress in the 'quest' based on completion of daily quizzes.
7. The user can see high-scores from past 'quests'.

Leila Stories

Tuesday, 25 March 2025 12:05 PM

Uni students:

1. No time
2. Forgetful (need the end date)
3. Appreciates aesthetics
4. Gets overwhelmed by lots of content - struggles to make flashcards, would appreciate it summarised

Requirement	User Stories	Acceptance Criteria
3. The user can use the app to generate flashcards from studied content.	As a busy university student with little free time, I want to generate flashcards quickly and easily, so that I can spend less time preparing to study.	Given there is an <u>input</u> in the application, when I paste my course content into the <u>input</u> , then the application will generate flashcards for me.
2. The user can specify an end date for each quest.	As a forgetful university student, I want to set an end date for each of my 'quests' which reflects my courses' final exam dates, so I won't forget when each of my final exams are.	Given multiple 'quests' can be created, when I create one of these quests, then I will be able to set an end date for it.
? The application is overall organised and aesthetically pleasing.	As an organised university student who prefers to study with resources that are aesthetically pleasing, I want the application to be <u>aesthetically organised</u> , so that my study can be more engaging and therefore effective.	Given the application is implemented, when I use it to study my course contents, then it will provide an organised interface to do so.
4. The user can [...] review flashcards.	As an easily overwhelmed university student, I want to view flashcards which summarise course content, so I can understand the content without being overwhelmed.	Given flashcards can be generated, when I generate these flashcards, then I will be able to review these flashcards at any time.
1. The user can create multiple 'quests'.	As an organised university student, I want to separate my flashcards for each of my courses, so that my study can be more organised and therefore effective.	Given flashcards and quizzes can be generated, when I start generating flashcards, then I will be able to separate them into different 'quests' for each of my courses.
4. The user can add [...] flashcards.	As a forgetful university student, I want to add more flashcards to quests over time, so I can add content I previously forgot about.	Given there is an <u>input</u> in the application and flashcards are already generated, when I paste more course content into the <u>input</u> , then the application will generate additional flashcards for my quest.
4. The user can edit [...] flashcards.	As a university student, I want to edit generated flashcards, so I can make sure they are right for me after being generated.	Given flashcards are already generated, when I select a flashcard, then I will be able to edit it.
4. The user can delete [...] flashcards.	As a university student, I want to delete generated flashcards, so I can make sure they correctly reflect course content if certain topics are irrelevant.	Given flashcards are already generated, when I select a flashcard, then I will be able to delete it.
1. The user can create multiple 'quests'.	As a student, I want to select a different character to represent each of my subjects, so I can more easily differentiate between each unit's content.	Given different 'quests' can be created for each of the user's subjects, when I create a new quest, then I will be able to select a character for that quest.
5. The user can complete daily quizzes based on generated flashcards.	As a competitive student, I want to earn rewards such as in-game items for completing daily quizzes, so I can have more of an incentive to keep studying.	Given users can complete daily quizzes, when I complete each daily quiz, then I can receive an in-app reward.
6. The user can progress in the 'quest' based on completion of daily quizzes.	As a competitive student, I want to see my progress on a completion bar, so I can study more due to the incentive to progress further.	Given users can progress in 'quests' based on completion of daily quizzes, when I progress in my quests, then I will be able to see my progression on a progress bar.
8. A graphical user interface with windows for the main functionalities, in general based on JavaFX	As a student, I want to study using an app with an interface (?), so I can study more easily.	Given there is an app, when I open the app, then it will display a graphical user interface for me to interact with.
9. An authentication system to sign-up/sign-in (GUI and models)	As a student, I want to sign up for an account in the app, so my information can be saved to an account.	Given there is a sign-up button, when I click this button, then I can enter my details to create my account.
	As a student, I want to sign into the app, so I can see information saved to my account.	Given there is a sign-in page, when I enter my account details, then I can use the app signed into my account.
10. A persistency system to store user data (GUI and models)	As a student, I want to store my data, so I don't lose my account after closing and reopening the app.	Given users can create accounts in the app, when I make my account, then the app will store my data.
11. a persistency system to retrieve user data (GUI and models)	As a student, I want to retrieve my data, so that I can see my past flashcards and progress.	Given users can create accounts in the app, when I log into my account, then the app will retrieve all my account data.
12. a persistency system to update user data (GUI and models)	As a student, I want to update my stored data, so that I can update my past flashcards and progress.	Given users can create accounts in the app, when I make updates in my account such as generating new flashcards, then the app will store them.
13. one or (more likely) more application windows in which the actual, useful part of the application is performed (again, with GUI and models).	As a student, I want to use a study app that has multiple different windows, so my study platform can be organised.	Given users can study on the application, when I use the app, then I will see multiple different windows implemented.

Requirements:

1. The user can create multiple 'quests'.
2. The user can specify an end date for each quest.
3. The user can use the app to generate flashcards from studied content.
4. The user can add, edit, delete and review flashcards.
5. The user can complete daily quizzes based on the generated flashcards.
6. The user can progress in the 'quest' based on completion of daily quizzes.
7. The user can see high-scores from past 'quests'.

A good user story will:

- Detail a functionality that responds to one or more requirements

Examples:

Requirements for Website catalog content management

- Create, update, publish and unpublish product catalog entries.
- Track all catalog changes.
- View and Edit product and product information including price
- View and edit all equipment hire, contract and information including price
- Create and/or Edit Customer details – General, Tradesperson or Business
- Create and/or Edit Customer order - list of products and quantities
- View Customer order – order, invoice, payment, receipt and tracking history
- Make a Supplier order - list of products and quantities, availability, receive invoice, make payment, receive receipt, receive goods, and restock inventory
- Schedule deliveries of customer orders. Schedule drop off or pickup of equipment hire.
- Record staff details, leave taken, and hours worked. Create payroll payment list for accountant. Create payment statements for staff for each payroll.
- Information System processes

Users: General, Business, Tradesperson

User Stories:

As a business user

I want to edit customer details

So that I can consult the database when needed.

(BAD - too technical talking about the database)

As a business user

I want to edit product details

So that I can update product information

As a tradesperson

I want to view the product catalogue

So that I know the cost of the product I want.

Should be:

Independent

Negotiable

Valuable

Estimable (required effort to implement)

Short & Simple (non-technical, and should not combine more than one requirement (e.g. cannot say I want to view and edit the catalogue in one)

Acceptance Criteria:

Given that there is a catalogue tag on the home page & correct information

When I click on the tag

Then I can view product information & price

- Be written from the perspective of a user of the system
- Include a clear goal and testable acceptance criteria

User Story Template

As a (e.g. Customer, User, Supplier, ...)
I want a (user requirement, product, feature)
so that (user benefit, value)

Acceptance Criteria

Given (pre-condition)
When (Action)
Then (Outcome)

Week 9 Submission Resources

Monday, 28 April 2025 10:21 AM

Design Video Script Week 9

Monday, 28 April 2025 10:10 AM

Walkthrough video contents:

Welcome to the primary prototype video for the CAB302 software development project.

- Explain the benefit of the project

The designed application for this is called AlfredAI. The overall purpose of this application is to assist students with studying efficiently by gamifying this process. In order to increase productivity by displaying engaging visuals and incentives to encourage continued study.

The structure of this gaming study tool is as follows each unit of study the user wants to complete is separated into different quests. The application generates flashcards based off notes submitted into the application or created by the user. These flashcards are then used to generate questions for daily quizzes. As you complete these quizzes your progress further in the quest. The length and end date of these quest is determined by the when the user needs to have mastered all of its material. The progression rate of the quest is dependent on the users exam scores and daily activity.

The overall benefit of structuring this study material and I gameable and interactive format is to increase the users drive to study. As the common causes for lack of study have stems from one of these possible underling issues;

1. Content being overwhelming due to either complexity or quantity of material
2. The inability to stay engaged with study as often they often get distracted or are prone to getting bored easily
3. The inability to study effectively, in terms of struggling with the generation of flashcards based off their notes, or they are missing content as they were in a rush to complete their study material
4. The lack of time due to commitments or limited time management skills

So, our application was designed to limit the impact these issues have on the users ability to study effectively and efficiently. Point 1, 3 and 4 is combated by having the possibility of AI creating the flashcards based off unit material and provided notes. While, 2 is achieved purely through the gamifying of the user experience.

- Overview of user stories and prioritisation

The user stories were created based on the requirement provided for this project as well as requirements generated by the project team. The content within the user stories follows the “As a”, “I want a” and “so that” structure, which outlines a given desire wanted by different user types. All of these have a listed acceptance criteria that explains what function in our design achieves this desire. Each user story has been allocated a prioritisation level.

“Must have” include the provided project requirements and core functions of our application and the project would fail if not included or would limit the ability to achieve benefits of this design.

“Should haves” are desires we have considered important however they are not considered vital for the first delivery of this project.

“Could haves” are function we would want in our application in order for it to preform to the highest level, however, will only be completed if time allows for there implementation.

Finally, “won’t haves” are not planned to completed within this release.

The requirements for this project are as follows;

- The user can create multiple 'quests'.
- The user can specify an end date for each quest.
- The user can use the app to generate flashcards from studied content.
- The user can add, edit, delete and review flashcards.
- The user can complete daily quizzes based on the generated flashcards.
- The user can progress in the 'quest' based on completion of daily quizzes.
- The user can see high-scores from past 'quests'.
- A graphical user interface with windows for the main functionalities, in general based on JavaFX.
- An authentication system to sign-up/sign-in.
- A persistency system to store user data.
- A persistency system to retrieve user data.
- A persistency system to update user data.
- One or more application windows in which the actual, useful part of the application is performed.

- Low fidelity prototypes of the User Interface

For this project three types of prototypes levels were created low, medium and a medium with the implementation of colours and images (*He called this something and I have forgotten feel free to change that wording*).

These are the low fidelity prototype, a rough sketch of the basic design layout in a wireframe like structure were created.

For the medium level prototypes the low fidelity sketches are transformed into clean and easily readable and were used to test user flows and interactions while providing the basis for the final design outcome.

The main user interfaces in this design consist of the following interfaces login, register, home, quest home, flashcards and quiz. The loading screen that also acts as the login in page is interface 1. The register screen that is accessed via the login page is interface 2. The creation of a new quest as outlined in 3 and 4 consists of a character section, quest information and selection for quest end date. Screen 5 shows the quest home one its selected where the flashcard and daily quiz can be accessed. The following screen shows the edit flashcards. Finally screens 7 and 8 show how the quiz is to be displayed and how the final results are displayed to the user.

This is the final prototype created before the initialisation of the coding process. Where the inclusion of images, backgrounds and icons that will be included in the final spirit of this project.

- Overview of Github and PM tool, including discussion of assigned tasks

For the completion of this project, a GitHub repository was used as a way for all project members to collaborate collectively on the code for this project with the use of coding within separate branches to then later push to the main repository and branch. As GitHub was being used as a basis for the code it was collaboratively decided that GitHub will be used as the project management tool for this project as well by using the project function that links to the repository.

including discussion of assigned tasks --> probs need more on this what I have done is below

The process of assigning tasks was carried out when all project members were present in our group meetings. The allocation was determined by what each member believed they were able to contribute to between weeks and meetings based off their commitments. Here are the meeting minutes taken for the meetings so far to help give an indication of how tasks are allocated. Also, the breakdown of coding tasks was separated by user stories and based off their priority level they were allocated into either sprint 1, sprint 2 or those that will be completed if time allows it. In order to achieve a consistent git log a git procedure was created for all members to follow to ensure that the list of GitHub commits accurately display the divided work log.

- Brief overview of the Java code created so far with emphasis on
 - JavaFX user interface

The first main screen is the application's welcome and login page. The main element of this page are a login form with a username and password input and a button to take users to the register page. This page is very similar to the login page except that during registration a confirmation of password is required. If either the login or register pages are successful it takes you to the main quest page.

The quest are displayed in a list format. This page gives you access to all the user's quests and allows the creation of new quests. Once a quest is selected the ability to access the daily quiz and flash cards are accessible. The main GUI elements in this page is the flipable flashcards to simulate the flipping of a flashcard in real life. Also the main interactive element in the exam interface is the progression bar that fills as users complete questions in the exam.

- Persistence (DB)

The database is set up into five tables. Users, user quests, quests, quest flashcards and flashcards. The user table consists of the userID that is auto incremented when a user registers, username and password. The user quest table is used to connect and bridge information between the user and quests table, this is a similar instance of the quest flashcards linking the Primary keys questID and flashcardsID. This is for every quest there is a list of flashcards and creates an easy connection between these tables.

The CRUD operations used are insert in the register, add quest and later add flashcard functions. Select in a display functions including the display of quests, flashcards and exam questions and answer. Update is mainly used in the edit flashcard functions. And finally, delete in the delete flashcard function.

- **Unit Tests**

In order to ensure code quality of the application unit test were performed on key methods to ensure methods are running as intended and returning the correct result. During the creation process, of these unit test a separate array of data is used in order to ensure that when testing the functions that would upload information to the database there is no complications when the test is run a second time.

These cases ensure code quality when it tests specific functions or methods as it isolates problems and checks edge cases. Allowing us to catch bugs early and improve code quality in order to achieve a bug free and smooth application.

- Additionally, for excellent marks
 - include a 1 minute demo showing the project in action. Demonstrate use of the functionalities implemented (preliminary GUI and DB)

Welcome to the video demonstration of our application AlfredAI

Welcome: This is the welcome and login in screen however if they don't have an account they can go to the register page.

Register: this is to the register page, in order to register an account all fields must be filled ***press enter no field filled*** an account must not exists an error is sent to the user ***error user exists***. Passwords must match ***passwords don't match***. And if all these succeed the user is registered ***succes*** once the ok button is pressed it takes you to the quest page ***next page***.

Quests: the quest page loads as a blank list until a quest has been created. To create a quest u need to follow the following props provided ***create quest*** these are name and end date and in the future character as well ***create quest confirm***. Once a quest is created flashcards and daily exams can be completed. The creation of flashcards will be created in the next sprint as the primary way this is done is through AI and the option for manual creation will also be added. In order to see the rest of the functions we will login to a premade account with this function manually completed.

Welcome (login to account with data): in order for login to be successfully the entered data must match the database otherwise an error message is displayed ***wrong password*** however if registered the user is able to login ***login successfully***.

Quests: once a quest is selected the user is able to access flashcards and exams.

Flashcards: this is the flashcards screen to see the answer on the flashcard the flip button is pressed the answer is shown ***press flip***. To see the next flashcard press next ***press next***. If it is discovered the flashcard is wrong the edit flashcards are wrong the edit flashcard can be pressed ***edit flashcards***.

Edit flashcards: this displays all flashcards in a list where they can either be deleted or be edited by the user. ***press edit*** In the edit function u can either edit the question or the answer. This will update the question or answer in the database. When deleting a flashcard in order to improve user experience a ***press delete*** confirmation button for deleting was added. However instead of a confirmation button yes needs to written by the user to confirm only then will it be deleted. ***press back buttons to go back to selected quest***

Quest: the other function was the exam. This exam is randomly generated from flashcards that have yet to masted by the user. When users start to answer questions the progression bar located at the top increments accordingly ***answer questions***. Once the users are happy with their results submit their answers they press submit ***press submit***. This will display the quiz results. This displays what was the correct answer and what u answer. Once they are finished looking at the results they are able to go back the quest page or home page to study more ***press continue***.

This also concludes the video demonstration for AlfredAI.

From <<https://d.docs.live.net/04758caf8294a697/Documents/Year%203%20Semester%201/CAB302/Script.docx>>

What was spoken

Friday, 2 May 2025 5:46 PM

Welcome to the primary prototype video for the CAB302 software development project.

The designed application for this is called AlfredAI. The overall purpose of this application is to assist students with studying efficiently by gamifying this process. In order to increase productivity by displaying engaging visuals and incentives to encourage continued study.

The structure of this gaming study tool is as follows each unit of study the user wants to complete is separated into different quests. The application generates flashcards based off notes submitted into the application or created by the user. These flashcards are then used to generate questions for daily quizzes.

As you complete these quizzes your progress further in the quest. The length and end date of these quest is determined by the ended specified by the user or when all the material is mastered. The progression rate of the quest is dependent on the users exam scores and daily activity.

The overall benefit of structuring this study material into a gameable and interactive format is to increase the users drive to study. As the common causes for lack of study stems from one of these possible underling issues;

1. Content being overwhelming due to either complexity or quantity of material
2. The inability to stay engaged with study as often they often get distracted or a prone to getting bored easily
3. The inability to study effectively, in terms of struggling with the generation of flashcards based off their notes, or they are missing content as they were in a rush to complete their study material
4. The lack of time due to commitments or limited time management skills

Hence, our application was designed to limit these impact on the users ability to study effectively and efficiently. Point 1, 3 and 4 is combated by having the possibility of AI creating the flashcards based off unit material and provided notes. While, 2 is achieved purely through the gamifying of the user experience.

The user stories were created based on the requirements provided for this project as well as requirements generated by the project team. The content within the user stories follows the "As a", "I want a" and "so that" structure, which outlines a given desire wanted by different user types. All of these have a listed acceptance criteria that explains what function in our design achieves this desire. And Each user story has been allocated a prioritisation level.

"Must haves" include the provided project requirements and core functions of our application. Without these the project would fail or would limit the ability to achieve benefits of this design.

"Should haves" are desires we have considered important however they are not considered vital for the first delivery of this project.

"Could haves" are function we would want in our application in order for it to preform to the highest level, however, will only be completed if time allows for there implementation.

Finally, "won't haves" are not planned to completed within this release.

The requirements for this project are as follows;

- The user can create multiple 'quests'.
- The user can specify an end date for each quest.
- The user can use the app to generate flashcards from studied content.
- The user can add, edit, delete and review flashcards.
- The user can complete daily quizzes based on the generated flashcards.
- The user can progress in the 'quest' based on completion of daily quizzes.
- The user can see high-scores from past 'quests'.
- A graphical user interface with windows for the main functionalities, in general based on JavaFX.
- An authentication system to sign-up/sign-in.
- A persistency system to store user data.
- A persistency system to retrieve user data.
- A persistency system to update user data.
- One or more application windows in which the actual, useful part of the application is performed.

For this project three types of prototypes levels were created low, medium and a medium with the implementation of colours and images.

These are the low fidelity prototype, a rough sketch of the basic design layout in a wireframe like structure were created.

For the medium level protypes the low fidelity sketches are transformed into clean and easily readable and were used to test user flows and interactions while providing the basis for the final design outcome.

The main user interfaces in this deign consist of the following interfaces login, register, home, quest home, flashcards and quiz. The loading screen that also acts as the login in page is interface 1.

The register screen that is accessed via the login page is interface 2.

The creation of a new quest as outlined in 3 and 4 consists of a character section, quest information and selection for quest end date.

Screen 5 shows the quest home where the flashcard and daily quiz can be accessed. The following screen shows the edit flashcards.

Finally screens 7 and 8 show how the quiz is to be displayed and how the final results are displayed to the user.

This is the final prototype created before the initialisation of the coding process. Where the inclusion of images, backgrounds and icons that will be included in the final sprit of this project.

For the completion of this project, a GitHub repository was used as a way for all project members to collaborate collectively on the code for this project with the use of coding within separate branches to then later push to the main repository and branch. As GitHub was being used as a basis for the code it was collaboratively decided that GitHub will be use as the project management tool for this project as well by using the project function that links to the repository.

The process of assign tasks was carried out when all project members wear present in our group meetings. The allocation was determined by what each member believed they wear able to contribute to between weeks and meetings based off their commitments.

These are the meeting meets taken so far to help give an indication of how tasks are allocated. Also, the breakdown of coding tasks was separated by user stories and based off their priority level they were allocated into either sprint 1, sprint 2. In order to achieve an accurate git log a git procedure was created for all members to follow to ensure that the list of GitHub commits accurately display the divided work log.

The first main screen is the applications welcome and login page. The main element of this page are a login form with a username and password input and a button to take users to the register page. This page is very similar to the login page accept that during registration a confirmation of password is required. If either the lock or register pages are successful it takes you to the main quest page.

The quest are displayed in a list for format. This area give you access to all the user quests and allow the

Welcome to the primary prototype video for the CAB302 software development project.

The designed application for this is called AlfredAI. The overall purpose of this application is to assist students with studying efficiently by gamifying this process. In order to increase productivity by displaying engaging visuals and incentives to encourage continued study.

The structure of this gaming study tool is as follows each unit of study the user wants to complete is separated into different quests. The application generates flashcards based off notes submitted into the application or created by the user. These flashcards are then used to generate questions for daily quizzes.

As you complete these quizzes your progress further in the quest. The length and end date of these quest is determined by the ended specified by the user or when all the material is mastered. The progression rate of the quest is dependent on the users exam scores and daily activity.

The overall benefit of structuring this study material into a gameable and interactive format is to increase the users drive to study. As the common causes for lack of study stems from one of these possible underling issues;

1. Content being overwhelming due to either complexity or quantity of material
2. The inability to stay engaged with study as often they often get distracted or a prone to getting bored easily
3. The inability to study effectively, in terms of struggling with the generation of flashcards based off their notes, or they are missing content as they were in a rush to complete their study material
4. The lack of time due to commitments or limited time management skills

Hence, our application was designed to limit these impact on the users ability to study effectively and efficiently. Point 1, 3 and 4 is combated by having the possibility of AI creating the flashcards based off unit material and provided notes. While, 2 is achieved purely through the gamifying of the user experience.

The user stories were created based on the requirements provided for this project as well as requirements generated by the project team. The content within the user stories follows the "As a", "I want a" and "so that" structure, which outlines a given desire wanted by different user types. All of these have a listed acceptance criteria that explains what function in our design achieves this desire. And Each user story has been allocated a prioritisation level.

"Must haves" include the provided project requirements and core functions of our application. Without these the project would fail or would limit the ability to achieve benefits of this design.

"Should haves" are desires we have considered important however they are not considered vital for the first delivery of this project.

"Could haves" are function we would want in our application in order for it to preform to the highest level, however, will only be completed if time allows for there implementation.

Finally, "won't haves" are not planned to completed within this release.

The requirements for this project are as follows;

- The user can create multiple 'quests'.
- The user can specify an end date for each quest.
- The user can use the app to generate flashcards from studied content.
- The user can add, edit, delete and review flashcards.
- The user can complete daily quizzes based on the generated flashcards.
- The user can progress in the 'quest' based on completion of daily quizzes.
- The user can see high-scores from past 'quests'.
- A graphical user interface with windows for the main functionalities, in general based on JavaFX.
- An authentication system to sign-up/sign-in.
- A persistency system to store user data.
- A persistency system to retrieve user data.
- A persistency system to update user data.
- One or more application windows in which the actual, useful part of the application is performed.

For this project three types of prototypes levels were created low, medium and a medium with the implementation of colours and images).

These are the low fidelity prototype, a rough sketch of the basic design layout in a wireframe like structure were created.

For the medium level protypes the low fidelity sketches are transformed into clean and easily readable and were used to test user flows and interactions while providing the basis for the final design outcome.

The main user interfaces in this deign consist of the following interfaces login, register, home, quest home, flashcards and quiz. The loading screen that also acts as the login in page is interface 1.

The register screen that is accessed via the login page is interface 2.

The creation of a new quest as outlined in 3 and 4 consists of a character section, quest information and selection for quest end date.

Screen 5 shows the quest home where the flashcard and daily quiz can be accessed. The following screen shows the edit flashcards.

Finally screens 7 and 8 show how the quiz is to be displayed and how the final results are displayed to the user.

The first main screen is the applications welcome and login page. The main element of this page are a login form with a username and password input and a button to take users to the register page. This page is very similar to the login page accept that during registration a confirmation of password is required. If either the lock or register pages are successful it takes you to the main quest page.

The quest are displayed in a list for format. This page gives you access to all the users quests and allows the creation of new quests. Once a quest is selected the ability to access the daily quiz and flash cards are accessible. The main QUI elements in this page is the flipable flashcards to simulate the flipping of a flashcard in real life. Also the main interactive element in the exam interface is the progression bar the fills as users complete questions in the exam.

The database is set up into five tables. Users, user quests, quests, quest flashcards and flashcards. The user table consist of the userID that is auto incremented when a user registers, username and password. The user quest table is used to connect and bridge information between the user and quests table, this is a similar instance of the quest flashcards linking the Primary keys questID and flashcardsID. This is for every quest there is lost of flashcards and creates an easy connection between these tables.

The CRUD operations used are insert ... in the register, add quest and later add flashcard functions. Select is used in all display functions such as the display of quests, flashcards and exam questions and answer. Update is mainly used in the edit flashcard functions. And finally, delete in the delete flashcard function.

In order to ensure code quality of the application unit test were performed on key methods to ensure methods are running as intended and returning the correct result. During the creation process, of these unit test a separate array of data is used in order to ensure that when testing the functions that would upload information to the database there is no complications when the test is run a second time.

These cases ensure code quality when it tests specific functions or methods as it isolate problems and checks edge cases. Allowing, us to catch bugs early and improve code quality in order to achieve a bug free and smooth application.

Information and selection for quest end date.

Screen 5 shows the quest home where the flashcard and daily quiz can be accessed. The following screen shows the edit flashcards.

Finally screens 7 and 8 show how the quiz is to be displayed and how the final results are displayed to the user.

This is the final prototype created before the initialisation of the coding process. Where the inclusion of images, backgrounds and icons that will be included in the final spirit of this project.

For the completion of this project, a GitHub repository was used as a way for all project members to collaborate collectively on the code for this project with the use of coding within separate branches to then later push to the main repository and branch. As GitHub was being used as a basis for the code it was collaboratively decided that GitHub will be use as the project management tool for this project as well by using the project function that links to the repository.

The process of assign tasks was carried out when all project members wear present in our group meetings. The allocation was determined by what each member believed they wear able to contribute to between weeks and meetings based off their commitments.

These are the meeting meets taken so far to help give an indication of how tasks are allocated. Also, the breakdown of coding tasks was separated by user stories and based off their priority level they were allocated into either sprint 1, sprint 2. In order to achieve an accurate git log a git procedure was created for all members to follow to ensure that the list of GitHub commits accurately display the divided work log.

The first main screen is the applications welcome and login page. The main element of this page are a login form with a username and password input and a button to take users to the register page. This page is very similar to the login page accept that during registration a confirmation of password is required. If either the lock or register pages are successful it takes you to the main quest page.

The quest are displayed in a list for format. This page gives you access to all the users quests and allows the creation of new quests. Once a quest is selected the ability to access the daily quiz and flash cards are accessible. The main QUI elements in this page is the flipable flashcards to simulate the flipping of a flashcard in real life. Also the main interactive element in the exam interface is the progression bar the fills as users complete questions in the exam.

The database is set up into five tables. Users, user quests, quests, quest flashcards and flashcards. The user table consist of the userID that is auto incremented when a user registers, username and password. The user quest table is used to connect and bridge information between the user and quests table, this is a similar instance of the quest flashcards linking the Primary keys questID and flashcardsID. This is for every quest there is lost of flashcards and creates an easy connection between these tables.

The CRUD operations used are insert ... in the register, add quest and later add flashcard functions. Select is used in all display functions such as the display of quests, flashcards and exam questions and answer. Update is mainly used in the edit flashcard functions. And finally, delete in the delete flashcard function.

In order to ensure code quality of the application unit test were performed on key methods to ensure methods are running as intended and returning the correct result. During the creation process, of these unit test a separate array of data is used in order to ensure that when testing the functions that would upload information to the database there is no complications when the test is run a second time.

These cases ensure code quality when it tests specific functions or methods as it isolate problems and checks edge cases. Allowing, us to catch bugs early and improve code quality in order to achieve a bug free and smooth application.

The video demonstration of Alfred AI will now begin.

Welcome: This is the welcome and login in screen however if they don't have an account they can go to the register page.

Register: this is to the register page, in order to register an account all fields must be filled *press enter no field filled* an account must not exists an error is sent to the user *error user exists*. Passwords must match *passwords don't match*. And if all these succeed the user is registered *success* once the ok button is pressed it takes you to the quest page *next page*.

Quests: the quest page loads as a blank list until a quest has been created. To create a quest u need to follow the following props provided *create quest* these are name and end date and in the future character as well *create quest confirm*. Once a quest is created flashcards and daily exams can be completed. The creation of flashcards will be created in the next sprint as the primary way this is done is through AI and the option for manual creation will also be added. In order to see the rest of the functions we will login to a premade account with this function manually completed.

Welcome (login to account with data): in order for login to be successfully the entered data must match the database otherwise an error message is displayed *wrong password* however if registered the user is able to login *login successfully*.

Quests: once a quest is selected the user is able to access flashcards and exams.

Flashcards: this is the flashcards screen to see the answer on the flashcard the flip button is pressed the answer is shown *press flip*. To see the next flashcard press next *press next*. If it is discovered the flashcard is wrong the edit flashcard can be pressed *edit flashcards*.

Edit flashcards: this displays all flashcards in a list where they can either be deleted or be edited by the user. *press edit* In the edit function u can either edit the question or the answer. This will update the question or answer in the database. When deleting a flashcard in order to improve user experience a *press delete* confirmation button for deleting was added. However instead of a confirmation button yes needs to be written by the user to confirm only then will it be deleted. *press back buttons to go back to selected quest*

Quest: the other function was the exam. This exam is randomly generated from flashcards that have yet to mastered by the user. When users start to answer questions the progression bar located at the top increments accordingly *answer questions*. Once the users are happy with their answers they press submit *press submit*. This will display the quiz results. This displays what the user has answered and the correct answer.

Once they are finished looking at the results they are able to go back the quest page or home page to study more *press continue*.

This also concludes the video demonstration for AlfredAI.

The video demonstration of Alfred AI will now begin.

Welcome: This is the welcome and login in screen however if they don't have an account they can go to the register page.

Register: this is to the register page, in order to register an account all fields must be filled *press enter no field filled* an account must not exists an error is sent to the user *error user exists*. Passwords must match *passwords don't match*. And if all these succeed the user is registered *success* once the ok button is pressed it takes you to the quest page *next page*.

Quests: the quest page loads as a blank list until a quest has been created. To create a quest u need to follow the following props provided *create quest* these are name and end date and in the future character as well *create quest confirm*. Once a quest is created flashcards and daily exams can be completed. The creation of flashcards will be created in the next sprint as the primary way this is done is through AI and the option for manual creation will also be added. In order to see the rest of the functions we will login to a premade account with this function manually completed.

Welcome (login to account with data): in order for login to be successfully the entered data must match the database otherwise an error message is displayed *wrong password* however if registered the user is able to login *login successfully*.

Quests: once a quest is selected the user is able to access flashcards and exams.

Flashcards: this is the flashcards screen to see the answer on the flashcard the flip button is pressed the answer is shown *press flip*. To see the next flashcard press next *press next*. If it is discovered the flashcard is wrong the edit flashcard can be pressed *edit flashcards*.

Edit flashcards: this displays all flashcards in a list where they can either be deleted or be edited by the user. *press edit* In the edit function u can either edit the question or the answer. This will update the question or answer in the database. When deleting a flashcard in order to improve user experience a *press delete* confirmation button for deleting was added. However instead of a confirmation button yes needs to be written by the user to confirm only then will it be deleted. *press back buttons to go back to selected quest*

Quest: the other function was the exam. This exam is randomly generated from flashcards that have yet to mastered by the user. When users start to answer questions the progression bar located at the top increments accordingly *answer questions*. Once the users are happy with their answers they press submit *press submit*. This will display the quiz results. This displays what the user has answered and the correct answer.

Once they are finished looking at the results they are able to go back the quest page or home page to study more *press continue*.

What is needed for Project Progress and Performance (Submission 1 - week 9 - 20 points)

Monday, 28 April 2025 10:19 AM

To complete your Project Progress and Performance assignment, you must create: (1) an archive (zipfile) of your project and (2) a video walkthrough of all the project files, sources, and component, plus a demo of the project in action.

Submit the video and zipfile in Canvas.

For more on how to format your submission files, see [Assessment 1: Description - Project Progress and Performance](#)

Milestone: 4 - Preliminary Project Submission

Worth points: 20/50

Project Requirements

At this stage in the semester, your project must include:

- *Preliminary (functional) Prototype: The project has a functional user interface and stores data to a DB for persistence*
- *Test-Driven Development: the project includes a suitable test suite*
- *Version Control Workflow: There is evidence of regular commits to repositories and proper source management practices (e.g. branching)*
- *Object Oriented Design: Initial use of Object Oriented design patterns(e.g. MVC) and principles (e.g. Encapsulation, Inheritance, ...)*

What we assess

We will assess your progress so far towards the unit learning outcomes related to this assignment:

1. Manage a software development project using an agile approach

Evidence of learning includes (not limited to):

- high quality agile artefacts (e.g. user stories, sprint plans)
- effective use of PM tools (e.g. Trello)
- effective use of CI/CD tools

2. Apply industry best practice tools and techniques including object-oriented design and test-driven development and continuous integration

Evidence of learning includes (not limited to):

- evidence of iterative design (e.g. sketches, wireframes)
- Evidence of TDD (e.g. unit tests)
- Evidence of effective code managing (e.g. branching)

3. Apply high performing team concepts to proactively build an effective team

Evidence of learning includes (not limited to):

- evidence of regular, consistent contribution (e.g. commits)
- evidence of effective communication (e.g. meeting minutes, comms log)

4. Justify appropriateness of technical choices and designs using written and oral communication

Evidence of learning includes (not limited to):

- effective and convincing demo of the project in the current state
- well crafted video with good, concise script
- comprehensive presentation of all required deliverables/artefacts

How we assess

One or more markers will review your submission and will express an assessment and provide some feedback based on the rubric below. The marker(s) will download your project (zipfile) and will explore the contents while watching (and with the help of) your video walkthrough. For each assessment criteria (see besides, 'what we assess') the markers will look for evidence in your project submission (zipfile) and in your video. It is imperative to include each and every assessed items in your zipfile and also discuss these in the video. For example, include all user stories in your zipfile as word documents, or excel, or other format that suits the team, and then in the video discuss them briefly, show where they are in the project tree, etc.

Note: do not assume that your tutor will mark your project. Craft your submission so that any person knowledgeable of the unit's contents can easily make sense of, and assess your work.



Walkthrough video contents:

- explain the benefit of the project
- overview of user stories and prioritisation
- low fidelity prototypes of the User Interface
- overview of Github and PM tool, including discussion of assigned tasks
- Brief overview of the Java code created so far with emphasis on

- JavaFX user interface
- Persistence (DB)
- Unit Tests
- Additionally, for excellent marks
- include a 1 minute demo showing the project in action. Demonstrate use of the functionalities implemented (preliminary GUI and DB)

Rubric

CAB302 Assessment 1: Project Progress and Performance - Project Submission		
Criteria	Ratings	Points
Agile project management view longer description	<p>35 pts Full marks</p> <p>29.75 pts High Distinction Clear evidence of highly effective use of Agile methodology: User-Stories are of high quality, with sensible prioritisation, acceptance criteria, and evidence of revisions. Release and Sprint planning are carefully justified. Documented regular use of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) demonstrate high proficiency in agile project management. Evidence that practices such as Integrated Build are adopted.</p> <p>26.25 pts Distinction Clear evidence of effective use of Agile methodology: User-Stories are of high quality, with prioritisation and acceptance criteria. Release and Sprint planning are carefully justified. Documented regular use of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) demonstrates proficiency in agile project management. Evidence that practices such as Integrated Build are adopted.</p> <p>22.75 pts Credit Evidence of effective use of Agile methodology: User-Stories include prioritisation and acceptance criteria. Release and Sprint planning are provided and discussed. Documented use of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) demonstrates proficiency in agile project management, although evidence of regular use may be insufficient.</p> <p>17.5 pts Pass Evidence of use of Agile methodology: User-Stories are provided, but may not include prioritisation or acceptance criteria. Release and Sprint planning are provided and discussed. Evidence of adoption of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) demonstrates attempts at agile project management, although with little evidence of regular use.</p> <p>14 pts Marginal Fail Evidence of use of Agile methodology: Some User-Stories are provided, but the quality is poor and many use cases are not covered. Release and Sprint planning are not provided or are not convincing. Evidence of adoption of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) is lacking, with little or no evidence of regular use.</p> <p>8.75 pts Fail There is little evidence of formal project management, if a tool was used, its use was sporadic and ineffective, there is no or very little evidence of use of agile practices.</p> <p>7 pts Low Fail Attempts to present project management, but insufficient detail is provided to evaluate it.</p> <p>0 pts No submission received</p>	/ 35 pts
Object Oriented Design	35 pts	/ 35 pts

[view longer description](#)**Full marks****29.75 pts****High Distinction**

Clean, intuitive, and fully functional project, with a complete user interface, well designed user experience, based on clear, realistic, and well crafted requirements. The requirements are implemented at the highest quality, with evidence of Test Driven Development (TDD), adoption, of coding standards, descriptive comments, and evidence of iterative design.

26.25 pts**Distinction**

Clean, intuitive, and mostly functional project, with a well designed user interface, based on clear, realistic requirements. The requirements are implemented at high quality, with evidence of Test Driven Development (TDD), comments and coding standards .

22.75 pts**Credit**

Clean, intuitive, and mostly functional project, with a well designed user interface. The requirements are implemented at a good standard, with evidence of unit testing and debugging.

17.5 pts**Pass**

The project is in a preliminary stage, not yet fully functional, with many feature still being implemented or tested. Persistence and User Interface may be not yet functioning (e.g. mockups).

14 pts**Marginal Fail**

The project is in a very preliminary stage, mostly not yet functional, with most features still being implemented or tested. Persistence and User Interface may be not yet functioning (e.g. mockups).

8.75 pts**Fail**

Not much code was implemented, and/or the quality does not demonstrate understanding of object oriented design principles, patterns, or application of good coding standards.

7 pts**Low Fail**

Attempts to present object oriented design, but insufficient detail is provided to evaluate it.

0 pts**No submission received**

Teamwork

[view longer description](#)**15 pts****Full marks**

/ 15 pts

12.75 pts**High Distinction**

The team is performing at the highest level. This is evidenced by adoption and regular, highly effective use of PM tools and practices, regular contribution by all team members, evidence of engagement and participation (e.g. through minutes of meetings, iterative improvement on artefacts)

11.25 pts**Distinction**

The team is performing at high level. This is evidenced by adoption and regular use of PM tools and practices, regular contribution by all team members, evidence of engagement and participation (e.g. through minutes of meetings, iterative improvement on artefacts)

9.75 pts**Credit**

The team is performing well. This is evidenced by adoption and use of PM tools and practices, contribution by all team members, evidence of engagement and participation (e.g. through minutes of meetings, iterative improvement on artefacts).

7.5 pts**Pass**

Teamwork Tools and Practices (e.g. regular meetings, PM tool, team contract) are adopted and documented in the video walkthrough and submitted material.

6 pts

	<p>Marginal Fail There is evidence the team has considered teamwork tools and practices, but little evidence of regular use.</p> <p>3.75 pts Fail There is little or no evidence of team management practices</p> <p>3 pts Low Fail Attempts to present teamwork, but insufficient detail is provided to evaluate it.</p> <p>0 pts No submission received</p>	
Delivery view longer description	<p>15 pts Full marks</p> <p>12.75 pts High Distinction The video walkthrough is comprehensive and convincing, crafted to the highest standard. It includes a Demo of the current progress, showing a functional prototype in action. A viewer is able to understand how the project is structured, what is current progress in the development, the rationale behind technical choices, and that all required deliverables have indeed been submitted.</p> <p>11.25 pts Distinction The video walkthrough is convincing and crafted to a high standard. It includes a Demo of the current progress, showing a functional prototype in action. A viewer is able to understand how the project is structured, what is current progress in the development, and that all required deliverables have indeed been submitted. Some technical decisions may be listed with no much rationale given, but overall the presentation is complete and credible.</p> <p>9.75 pts Credit The video walkthrough is convincing and crafted to a good standard. It includes a Demo of the current progress, showing a prototype in action, though only some features are functional at the present stage. A viewer is able to understand how the project is structured, what is current progress in the development, although with some gaps or missing information. Some technical decisions may be listed with no much rationale given, but overall the presentation is credible.</p> <p>7.5 pts Pass The video walkthrough is crafted to a good standard. A viewer is able to understand how the project is structured, what is current progress in the development, although with some gaps or missing information. Some technical decisions may be listed with no much rationale given, but overall the presentation is credible.</p> <p>6 pts Marginal Fail The video walkthrough is rushed and incomplete. The project structured, and current progress in the development, are not clearly discussed. Technical decisions are listed with no rationale given, making the overall presentation hard to judge.</p> <p>3.75 pts Fail The video walkthrough does not effectively demonstrate the project's functionalities, a commentary is not provided, or is of very poor quality</p> <p>3 pts Low Fail Attempts to present a project, but insufficient detail is provided to evaluate it.</p> <p>0 pts No submission received</p>	/ 15 pts
Total points: 0		

Week 13 Submissions

Tuesday, 6 May 2025 8:51 PM

Submission 2 - week 13 - 30 points - Project Progress and Performance

Tuesday, 6 May 2025 8:48 PM

Details

To complete your Project Progress and Performance assignment, you must create: (1) an archive (zipfile) of your project and (2) a video walkthrough of all the project files, sources, and component, plus a demo of the project in action.

Submit the video and zipfile in Canvas.

For more on how to format your submission files, see [Assessment 1: Description - Project Progress and Performance](#)

Milestone: 6 - Finalised Project Submission

Worth points: 30/50

Project Requirements

At this stage in the semester, your project must include:

- *The Final Prototype delivers the [minimal requirements](#):*
- a graphical user interface with windows for the main functionalities, in general based on JavaFX
- an authentication system to sign-up/sign-in (GUI and models)
- a persistency system to store/retrieve/update user data (GUI and models)
- one or (more likely) more application windows in which the actual, useful part of the application is performed (again, with GUI and models).
- *All 'Must-Have' features have now been implemented as they were detailed in the user stories*
- *Build Script: Build and Tests are automated using a build script in Github*
- *Automated Build Server: Github 'Actions' are in place to automated the CI/CD process*
- *Documentation: Javadoc comments and generated pages.*
- *Object Oriented Design: Advanced use of Object Oriented design patterns (e.g. Dependency Injection, Builder, Singleton, Factory, Observer) and principles (abstraction, encapsulation, inheritance, and polymorphism)*

What we assess

We will assess your progress so far towards the unit learning outcomes related to this assignment:

1. Manage a software development project using an agile approach

Evidence of learning includes (not limited to):

- high quality agile artefacts (e.g. user stories, sprint plans)
- effective use of PM tools (e.g. Trello)
- effective use of CI/CD tools

2. Apply industry best practice tools and techniques including object-oriented design and test-driven development and continuous integration

Evidence of learning includes (not limited to):

- evidence of iterative design (e.g. sketches, wireframes)
- Evidence of TDD (e.g. unit tests)
- Evidence of effective code managing (e.g. branching)

3. Apply high performing team concepts to proactively build an effective team

Evidence of learning includes (not limited to):

- evidence of regular, consistent contribution (e.g. commits)
- evidence of effective communication (e.g. meeting minutes, comms log)

4. Justify appropriateness of technical choices and designs using written and oral communication

Evidence of learning includes (not limited to):

- effective and convincing demo of the project in the current state
- well crafted video with good, concise script
- comprehensive presentation of all required deliverables/artefacts

How we assess

One or more markers will review your submission and will express an assessment and provide some feedback based on the rubric below. The marker(s) will download your project (zipfile) and will explore the contents while watching (and with the help of) your video walkthrough. For each assessment criteria (see besides, 'what we assess') the markers will look for evidence in your project submission (zipfile) and in your video. It is **imperative** to include each and every assessed items in your zipfile and also discuss these in the video. For example, include all user stories in your zipfile as word documents, or excel, or other format that suits the team, and then in the video discuss them briefly, show where they are in the project tree, etc.

Note: do not assume that your tutor will mark your project. Craft your submission so that any person knowledgeable of the unit's contents can easily make sense of, and assess your work.

Walkthrough video contents:

- Detailed presentation of the Java code created so far with emphasis on
 - JavaFX user interface
 - Persistence (DB)
 - Unit Tests
- Detailed presentation of the CI/CD and build automation scripts in Github
 - Github actions
 - Automated Tests/Builds
- Demo of the application in action
- Additionally, for excellent marks
 - Outcomes of the User Testing

View Rubric

CAB302	Ratings	Points
Criteria		
Agile project management view longer description	35 pts Full marks 29.75 pts High Distinction Clear evidence of highly effective use of Agile methodology: User-Stories are of high quality, with sensible prioritisation, acceptance criteria, and evidence of revisions. Release and Sprint planning are carefully justified. Documented regular use of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) demonstrate high proficiency in agile project management. Evidence that practices such as Integrated Build are adopted.	/ 35 pts
	26.25 pts Distinction Clear evidence of effective use of Agile methodology: User-Stories are of high quality, with prioritisation and acceptance criteria. Release and Sprint planning are carefully justified. Documented regular use of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) demonstrates proficiency in agile project management. Evidence that practices such as Integrated Build are adopted.	
	22.75 pts Credit Evidence of effective use of Agile methodology: User-Stories include prioritisation and acceptance criteria. Release and Sprint planning are provided and discussed. Documented use of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) demonstrates proficiency in agile project management, although evidence of regular use may be insufficient.	
	17.5 pts Pass Evidence of use of Agile methodology: User-Stories are provided, but may not include	

prioritisation or acceptance criteria. Release and Sprint planning are provided and discussed. Evidence of adoption of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) demonstrates attempts at agile project management, although with little evidence of regular use.

14 pts

Marginal Fail

Evidence of use of Agile methodology: Some User-Stories are provided, but the quality is poor and many use cases are not covered. Release and Sprint planning are not provided or are not convincing. Evidence of adoption of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) is lacking, with little or no evidence of regular use.

8.75 pts

Fail

There is little evidence of formal project management, if a tool was used, its use was sporadic and ineffective, there is no or very little evidence of use of agile practices.

7 pts

Low Fail

Attempts to present project management, but insufficient detail is provided to evaluate it.

0 pts

No submission received

Object Oriented Design

[view longer description](#)

35 pts

Full marks

/ 35 pts

29.75 pts

High Distinction

Clean, intuitive, and fully functional project, with a complete user interface, well designed user experience, based on clear, realistic, and well crafted requirements. The requirements are implemented at the highest quality, with evidence of Test Driven Development (TDD), adoption, of coding standards, descriptive comments, and evidence of iterative design.

26.25 pts

Distinction

Clean, intuitive, and mostly functional project, with a well designed user interface, based on clear, realistic requirements. The requirements are implemented at high quality, with evidence of Test Driven Development (TDD), comments and coding standards .

22.75 pts

Credit

Clean, intuitive, and mostly functional project, with a well designed user interface. The requirements are implemented at a good standard, with evidence of unit testing and debugging.

17.5 pts

Pass

The project is in a preliminary stage, not yet fully functional, with many feature still being implemented or tested. Persistence and User Interface may be not yet functioning (e.g. mockups).

14 pts

Marginal Fail

The project is in a very preliminary stage, mostly not yet functional, with most features still being implemented or tested. Persistence and User Interface may be not yet functioning (e.g. mockups).

8.75 pts

Fail

Not much code was implemented, and/or the quality does not demonstrate understanding of object oriented design principles, patterns, or application of good coding standards.

7 pts

Low Fail

Attempts to present object oriented design, but insufficient detail is provided to evaluate it.

	0 pts No submission received	
Teamwork view longer description	15 pts Full marks <p>12.75 pts High Distinction The team is performing at the highest level. This is evidenced by adoption and regular, highly effective use of PM tools and practices, regular contribution by all team members, evidence of engagement and participation (e.g. through minutes of meetings, iterative improvement on artefacts)</p> <p>11.25 pts Distinction The team is performing at high level. This is evidenced by adoption and regular use of PM tools and practices, regular contribution by all team members, evidence of engagement and participation (e.g. through minutes of meetings, iterative improvement on artefacts)</p> <p>9.75 pts Credit The team is performing well. This is evidenced by adoption and use of PM tools and practices, contribution by all team members, evidence of engagement and participation (e.g. through minutes of meetings, iterative improvement on artefacts).</p> <p>7.5 pts Pass Teamwork Tools and Practices (e.g. regular meetings, PM tool, team contract) are adopted and documented in the video walkthrough and submitted material.</p> <p>6 pts Marginal Fail There is evidence the team has considered teamwork tools and practices, but little evidence of regular use.</p> <p>3.75 pts Fail There is little or no evidence of team management practices</p> <p>3 pts Low Fail Attempts to present teamwork, but insufficient detail is provided to evaluate it.</p> <p>0 pts No submission received</p>	/ 15 pts
Delivery view longer description	15 pts Full marks <p>12.75 pts High Distinction The video walkthrough is comprehensive and convincing, crafted to the highest standard. It includes a Demo of the current progress, showing a functional prototype in action. A viewer is able to understand how the project is structured, what is current progress in the development, the rationale behind technical choices, and that all required deliverables have indeed been submitted.</p> <p>11.25 pts Distinction The video walkthrough is convincing and crafted to a high standard. It includes a Demo of the current progress, showing a functional prototype in action. A viewer is able to understand how the project is structured, what is current progress in the development, and that all required deliverables have indeed been submitted. Some technical decisions may be listed with no much rationale given, but overall the presentation is complete and credible.</p>	/ 15 pts

9.75 pts**Credit**

The video walkthrough is convincing and crafted to a good standard. It includes a Demo of the current progress, showing a prototype in action, though only some features are functional at the present stage. A viewer is able to understand how the project is structured, what is current progress in the development, although with some gaps or missing information. Some technical decisions may be listed with no much rationale given, but overall the presentation is credible.

7.5 pts**Pass**

The video walkthrough is crafted to a good standard. A viewer is able to understand how the project is structured, what is current progress in the development, although with some gaps or missing information. Some technical decisions may be listed with no much rationale given, but overall the presentation is credible.

6 pts**Marginal Fail**

The video walkthrough is rushed and incomplete. The project structured, and current progress in the development, are not clearly discussed. Technical decisions are listed with no rationale given, making the overall presentation hard to judge.

3.75 pts**Fail**

The video walkthrough does not effectively demonstrate the project's functionalities, a commentary is not provided, or is of very poor quality

3 pts**Low Fail**

Attempts to present a project, but insufficient detail is provided to evaluate it.

0 pts**No submission received**

CAB302 Assessment 1: Project Progress and Performance - Project Submission

From <<https://canvas.qut.edu.au/courses/20355/assignments/183434>>

Liv's Script for Final Video

Tuesday, 27 May 2025 8:42 PM

Let me know if there are questions



My Script
Section fo...



My Script
Section fo...

Welcome to the final presentation for CAB302 software development project.

Benefits

The designed application for this is called AlfredAI. The overall purpose of this application is to assist students with studying efficiently by gamifying this process. In order to increase productivity by displaying engaging visuals and incentives to encourage continued study.

Commented [OM1]: Benefits

The structure of this gaming study tool is as follows each unit of study the user wants to complete is separated into different quests. The application generates flashcards based off notes submitted into the application or created by the user. These flashcards are then used to generate questions for daily quizzes. As you complete these quizzes your progress further in the quest. The length and end date of these quest is determined by the when the user needs to have mastered all of its material. The progression rate of the quest is dependent on the users exam scores and daily activity.

The overall benefit of structuring this study material and I gameable and interactive format is to increase the users drive to study. As the common causes for lack of study stems from one of the displayed underling issues. **(1 Minute)**

Commented [OM2]: Slide with study issues

Commented [OM3R2]: 1.Content being overwhelming due to either complexity or quantity of material
2.The inability to stay engaged with study as often they often get distracted or a prone to getting bored easily
3.The inability to study effectively, in terms of struggling with the generation of flashcards based off their notes, or they are missing content as they were in a rush to complete their study material
4.The lack of time due to commitments or limited time management skills

Commented [OM4]: Slide with requirements broken into two separate categories

Requirements → pre-determined → group determined

In order to complete this project a list of requirements for the created application needed to be determined. There are two types of requirements for this project, the ones provided by the project outline that must be in the application for it to be considered as complete. The other type of requirements are ones generated by the members of the project team on what we want the application to achieve. These can be seen listed here.

Commented [OM5]: Slide with just sprint 1 on it

Sprint 1

To manage workload for this project the content both in terms of code and other activities were divided into fortnightly sprints that would be completed by all member of the project. The first sprint to took place from March 25th to April 8th this covered the initiation process of the project and key required steps. Including sections such as the creation of the UI interface and the completion of user stories that will outline our project.

User stories + prioritisation

The first completed step for this sprint was the user stories, each member of the group contributed minimum four user stories based on their requirements and user type that where then collated into a master list. The structure of these stories follows the provided "As a", "I want a" and "so that" structure. Here is some examples of created user stories. **(1 Minute)**

Prototypes

The second required element for this sprint was the completion of both low and two versions of medium fidelity UI interfaces. These are the low fidelity prototype, a rough sketch of the basic design layout in a wireframe-like structure were created.

Commented [OM6]: Low fidelity prototypes

For the medium level prototypes the low fidelity sketches are transformed into clean and easily readable and were used to test user flows and interactions while providing the basis for the final design outcome.

Commented [OM7]: Medium fidelity

The main interfaces include login, register, quest home, flashcards and quiz. The login interface also acts as the main home page of our application. The creation of a quest is displayed over two screens in these prototypes however in the final creation of the design this was incorporated into a singular interface screen. The quest home page allows access to the flashcard and daily quizzes for its corresponding quest. The edit flashcards allows users to edit both the question and answer of all listed flashcards. Finally, you have the exam interfaces for both the quiz and solutions.

The second type of medium fidelity prototypes created incorporated a possible layout including images and possible colour schemes. (1 Minute)

Commented [OM8]: Medium 2

Sprint 2
The second sprint took place from the 9th to the 29th of April. During this sprint the first level of code was to be completed incorporating all Must Haves determined in the user stories for both prescribed and self-determined requirements. Also, for the start of this sprint key elements such as the project management tool and the diffusion of assigning tasks was determined in our group meeting at start of this sprint.

Commented [OM9]: Sprint 2

PM tool
For the completion of this project, a GitHub repository was used as a way for all project members to collaborate collectively on the code for this project with the use of coding within separate branches to then later push to the main repository and branch. As GitHub was being used as a basis for the code it was collaboratively decided that GitHub will be used as the project management tool for this project as well by using the project function that links to the repository.

Commented [OM10]: PM Tool

The process of assign tasks was carried out when all project members were present in our group meetings as outlined in the task assigned in the sprint outline. These were determined based on team members commitments for the two weeks the sprint covered.

Mention git hub issue → git procedure

During this sprint it was discovered that each team member was using GitHub different in terms of how they pushed and when and overall how they used it. This caused large discrepancies in the amount of commits by each member hence, the git log did not accurately displaying the division of workload. (Just Over 1 Minute)

Therefore, at the end this sprint a GitHub procedure was created that all team members were required to follow. This meant going forward the amount of commits would more accurately show the division of workload. Except, when code was worked on collaboratively within group meetings.

Commented [OM11]: Github procedure

Little side note

The following three sprint do not have an new information or much key documents that can easily be displayed in this video. For more details and documentation refer to the OneNote extraction PDF.

Sprint 3

Sprint three ended on the 13th of May, covered the implementation of should haves and the remainder of must haves that directly related to the implementation of the AI Ollama. There were small implications in this sprint to do with inaccurately evaluating how much work specific elements in this sprint would take. However, these were overcome by recalculating work with group member who had more time.

Commented [OM12]: Sprint 3

Sprint 4

Sprint four ended on May 20th, this sprint mainly covered the finalisation of content for the in-person presentation that would take place on the 27th of May. As well as the finalization of code.

Commented [OM13]: Sprint 4

Sprint 5

Sprint five concluded on Friday the 30th of May. This sprint included all detail relating to the final submission due on the same date.

Commented [OM14]: Sprint 5

Meeting minutes

For all group meetings that took place during project detailed scrum or meeting minutes was taken to ensure the project process was accurately demonstrated. Here are some examples of these meetings minutes.

Commented [OM15]: Examples of group meetings

(Final with stops to catch breath 5 mins 20 seconds)

Leila's Script

Wednesday, 28 May 2025 11:10 AM

We implemented continuous integration through GitHub Actions, which allowed us to run automatic tests and builds (in the form of a ‘workflow’) every time a change was pushed to the main branch. Examples can be seen here, where workflows were run for every pull request and merge into main.

This was the start of the implementation, where you can see when we first got these automated tests working.

And here’s a good example of a workflow helping us detect errors we hadn’t noticed. Here, only minor edits were made but some of these involved slightly altering the quiz logic, which I hadn’t noticed had caused one of the previously-written unit tests to fail. Because of this automated testing, I was able to notice the error with my new logic and fix it before merging my branch with main.

(approx 40sec)

Emily's Script

Wednesday, 28 May 2025 11:23 AM

Starting with the JavaFX user interface, the UI is designed with multiple screens to facilitate the main functionalities. For example, the Login window uses JavaFX controls such as TextFields and PasswordFields, with event handlers that validate user input in real time, showing clear error messages if the credentials are invalid. As displayed on this slide. The Registration window similarly includes fields with validation logic to ensure correct and secure user data entry before allowing account creation.

After authentication, users access the main application window where features like viewing quests, managing flashcards and completing quizzes are implemented. These screens use JavaFX components like ListView and Buttons, and bind UI elements directly to data models to reflect changes immediately. The UI also incorporates fun, fantasy themed visuals to improve user engagement. By also implementing the concept of mastered flashcards user can easily navigate and study flashcards that are more beneficial to their study.

For persistence, we use a relational database to securely store user data, quests, and flashcards. The persistence layer is implemented via the Data Access Object (DAO) pattern, which separates database operations from UI and business logic. Each DAO class handles Create, Read, Update, and Delete operations using prepared SQL statements to prevent security risks like SQL injection.

To efficiently manage database connections, we implement a Singleton pattern in the database connection manager, ensuring there is a single shared connection resource, which optimises performance and resource usage.

Regarding unit tests, we follow Test-Driven Development principles using JUnit 5. Tests cover critical functionality such as authentication logic, establishment of the database connection, and quiz creation behaviour. For database-related tests, we use mock objects to isolate the logic and avoid dependencies on the actual database during testing.

Throughout development, we followed an iterative design process, refining the UI and backend logic based on regular user feedback and testing cycles to ensure continuous improvement. For example navigation buttons to each page were made easier to use based on this feedback.

Overall, the code architecture applies sound object-oriented principles, clean separation of concerns, and robust testing, ensuring the project is both maintainable and scalable. To support collaboration, all classes are fully documented with Javadocs, making the purpose and behaviour of our code easy to understand for other developers.

CRITERIA

Object Oriented Design:

- The requirements are implemented at the highest quality, with evidence of...
- Test Driven Development (TDD),
- adoption, of coding standards,
- descriptive comments, and
 - Maybe mention how Javadocs used for code documentation, so all classes are commented descriptively for other developers.
- evidence of iterative design.
- Really not sure how we talk about this...

Delivery:

- A viewer is able to understand how the project is structured,
- what is current progress in the development,
- the rationale behind technical choices, and
- that all required deliverables have indeed been submitted.

Maybe change second sentence to:

Tests cover critical functionality such as authentication logic, establishment of the database connection, and quiz creation behaviour.

Should the last sentence be this:

For example navigation buttons to each page were made easier to use based on this feedback.

Video

Tuesday, 20 May 2025 12:39 PM

Walkthrough video contents:

- Week 9 content - Liv
- Detailed presentation of the Java code created so far with emphasis on
 - JavaFX user interface
 - Persistence (DB)
 - Unit Tests - Emily & Leila
- Detailed presentation of the CI/CD and build automation scripts in Github
 - Github actions
 - Automated Tests/Builds - Leila SCREEN RECORDING
- Demo of the application in action - Max
- Additionally, for excellent marks
 - Outcomes of the User Testing - Dani

5min
1min
2min
2min
?

CRITERIA

Agile:

Clear evidence of highly effective use of Agile methodology:

- User-Stories are of high quality, with sensible prioritisation, acceptance criteria, and evidence of revisions.
- Release and Sprint planning are carefully justified.
- Documented regular use of project management tools (GitHub Projects, Microsoft Teams, Trello, etc.) demonstrate high proficiency in agile project management.
- Evidence that practices such as Integrated Build are adopted

Object Oriented Design:

- Clean, intuitive, and fully functional project, with a complete user interface, well designed user experience, based on clear, realistic, and well crafted requirements.
- The requirements are implemented at the highest quality, with evidence of Test Driven Development (TDD), adoption, of coding standards, descriptive comments, and evidence of iterative design.

Teamwork:

- The team is performing at the highest level. This is evidenced by adoption and regular, highly effective use of PM tools and practices, regular contribution by all team members, evidence of engagement and participation (e.g. through minutes of meetings, iterative improvement on artefacts)

Delivery:

- The video walkthrough is comprehensive and convincing, crafted to the highest standard.
- It includes a Demo of the current progress, showing a functional prototype in action.
- A viewer is able to understand how the project is structured, what is current progress in the development, the rationale behind technical choices, and that all required deliverables have indeed been submitted.

We will assess your progress so far towards the unit learning outcomes related to this assignment:

1. Manage a software development project using an agile approach

Evidence of learning includes (not limited to):

- high quality agile artefacts (e.g. user stories, sprint plans)
- effective use of PM tools (e.g. Trello)
- effective use of CI/CD tools

2. Apply industry best practice tools and techniques including object-oriented design and test-driven development and continuous integration

Evidence of learning includes (not limited to):

- evidence of iterative design (e.g. sketches, wireframes)
- Evidence of TDD (e.g. unit tests)
- Evidence of effective code managing (e.g. branching)

3. Apply high performing team concepts to proactively build an effective team

Evidence of learning includes (not limited to):

- evidence of regular, consistent contribution (e.g. commits)
- evidence of effective communication (e.g. meeting minutes, comms log)

4. Justify appropriateness of technical choices and designs using written and oral communication

Evidence of learning includes (not limited to):

- effective and convincing demo of the project in the current state
- well crafted video with good, concise script
- comprehensive presentation of all required deliverables/artefacts

From <<https://canvas.qut.edu.au/courses/20355/assignments/183434>>

Video Demo Outline (Max)

Tuesday, 13 May 2025 11:53 AM

Video Order --> for script and order

- Register user
 - o Show all invalid
 - Not filled
 - Passwords don't match
 - Already exists
- Add 2 subject
- Show 0 flashcards
- Generate 3 with AI
- Add 1 manually
- Flashcards flip and next
- Edit
- Delete
- Quiz
- Show result page
- All quest --> other quest
- Show no flashcards / no overlap
- Logout
- Log back in as user to both progress and login in function
 - o Incorrect password or username
 - o Success

Welcome to alfred AI, we're going to start by registering a user, the application checks if all filleds are filled, matching passwords and if a user already exists.

Once all these are valid it logs the user in. They will then be taken to the add subject menu, let's start by adding the first subject,

CAB302 with a due date of 31st of may, as well as my other class CAB203 with a due date of 9th of jun.

Let's go into our first one as this is due first, at the moment there are no flashcards, lets generate 3 with AI based on the first weeks content,

great but I also want to add one manually based on the content.

Now that we have our four flashcards lets take a look. Within this menu I am able to edit and delete the flash cards.

Finally with our three flashcards lets take a quiz. Inputting the results we get a bar of our progress

Now that we've completed our quest for the day, I can check the other subject to see check that there are no flash cards from the other subject

I'm going to log out. I can log back in, the program will check for a incorrect username/password, however on the correct submission I get logged in.

Now that I've used the program for two days my achievements will of updated. Lets take a look to check my progress. This has been a demo of alfred AI

1 Minute per slide MAXXXXXX

CAB302

FINAL PRESENTATION TEMPLATE (V-2025.02.13)

1. Project Overview

What it is, what it does, what it looks like

You should include a screenshot

List of key features as bullet points

Max

Script from week 9

The structure of this gaming study tool is as follows each unit of study the user wants to complete is separated into different quests. The application generates flashcards based off notes submitted into the application by the user. These flashcards are used to practice and study for the quest. There are daily quizzes for daily quizzes. As you complete these quizzes your progress further in the quest. The length and end date of these quest is determined by when the user needs to have mastered all of its material. The progression rate of the quest is dependent on the users exam scores and daily activity.

“

Hi we're AlfredAI I'm Max, this is Emily, Danielle, Oliva, and Lilea.

The purpose of AlfredAI is to assist students in studying more efficiently and effectively. This is done by gamifying the study process.

In order to make it more engaging we enhance the interactivity and provide incentives to encourage continued study.

The structure of the application is based that a user has a quest based on every subject they wish to study.

While there are many features in the application the core features are:

1. Gamification of studying such as bosses, achievements.
2. The usage of AI to generate flashcards based off users subjects
3. The ability to create daily quizzes based on the created flashcards

To encourage continue study we implemented:

1. An adventure game format and an
2. AI availability to generate flashcards

From: <https://id.docu.be.net/04D94a9f294a497/Documents/Year%201%20Semester%201%20%20Project%20Week%201%20Checklist%20Template.docx>

Possible key features:

1. Flashcard generation
2. Daily quizzes
3. Study gamification

SLIDE PRESENTED BY:

2. Project Planning and Design

Describe the planning process:

- How did you decide what to do?
- Did the plan work throughout the semester?
- What adaptations did you have to make?

You should include examples of requirements, user stories, other planning artefacts if any

Dani

The inspiration for the base of this project was taken from the idea of creating a study app. In terms of deciding the overall visuals and interactive method was inspired from our love for fantasy culture and was a key influence in creating aspects of the application.

To plan through the sprints we allocated a person's workload based on due dates provided by checklists, deadlines and corresponding sprints. The allocation of workload per person was determined by the individual commitments and what would be achievable by the end of the sprint.

Using this method allowed us to divide the workload evenly between members in both terms of code and non-code related task such as, project management, documentation and video preparation.

SLIDE PRESENTED BY:

3. Technical Details

Present in more details the technologies used in the system, motivate your technical decisions, list and comment on the supporting software/libraries you needed to use (if any)

You should include a visual representation of the main component, and how they interrelate

Lila

The three main technologies used in this project were GitHub, IntelliJ IDEA and OneNote.

GitHub was used as both a code collation software and a project management tool. IntelliJ with Amazon Corretto, git and Ollama AI was used as our coding platform, with the code then being pushed to GitHub. The Ollama LLM was used to implement the AI functionality, with the Llama 3 model. Finally, OneNote was used as a collection of work platform that contained documents and collections that is not yet fully developed. Examples of this include: the release plan, user stories, meeting minutes, sprint release plans, UI designs and database and code planning.

The main three programming languages used in the creation of this application are Java, JavaFX for front end development, and SQLite for database management and implementation. Our code follows the basic structure of FXML for front-end elements and controller classes for implementing the functionalities. (Diagram for this).

Unit tests were also written to test the functionality of the classes' methods, separate from the database.

Check if we need this part

The Amazon Corretto library is a software development tool, part of the Open Java Development Kit containing supporting libraries combined with the Java Runtime Environment and Java Virtual Machine .

“

From: <https://id.docu.be.net/04D94a9f294a497/Documents/Year%201%20Semester%201%20%20Project%20Week%201%20Checklist%20Template.docx>

SLIDE PRESENTED BY:

Criteria:

- The project is described in detail, including:
 - goals (what the project aims to achieve)
 - scope (what is included and decided to implement)
 - overall value (why the project/Application is needed)
- Features are listed and discussed exhaustively and convincingly
- The technical decisions made during the semester are convincing and reasonable
- There is evidence that different alternatives were considered, and this is provided through specific extracts of discussions and teamwork.
- The technical issues encountered are presented and contextualised, including explaining how these drove some technical decisions, or impacted the plan.
- Quality Assurance practices are clearly discussed, with details and rationale
- A reflection is presented on the relative merits and different applicability of different approaches, such as unit testing, test driven development, etc.
- Examples are provided of how testing was conducted, including to address bugs or other technical issues that arose.
- Technologies used in the system, motivate your technical decisions, list and comment on the supporting software/libraries you used

4. Project Quality

Detail your approach to ensuring project quality

- Agile Practices adopted
- Your reflections on what worked well for your project
- You should include one example

To ensure the quality and reliability of our application, we introduced a structured GitHub workflow. Early commit logs didn't accurately reflect codebase distribution due to file renames, so we implemented a shared process where each contribution was tracked through issues, branches, and pull requests. Improving accountability and user feedback to enhance maintainability.

We applied unit testing on core model classes—such as User, Quiz, and Database—to verify functionality and reduce regressions. We also refactored based on test outcomes and user feedback to enhance maintainability.

Agile practices were key to our planning. We worked in two-week sprints, held team meetings every week, discussed sprint timelines and sprint goals in shared minutes. As can be seen on screen, maintaining alignment and accountability throughout.

During Sprint 3, we identified a usability issue with the flashcard UI. Based on testing and user feedback, we redesigned the layout within the same sprint. Agile's iterative approach allowed us to respond quickly and improve user experience in real time.

SLIDE PRESENTED BY:

5. Teamwork

Discuss your approach for improving teamwork

- What tools did you adopt?
- What challenges did you encounter?
- What did you do in response?

You should include an episode from the semester of a challenging/conflictual situation, and how the team addressed this problem

Olivia

In order to achieve effective teamwork, we used GitHub for code collaboration and Discord for communication. A shared OneNote project to track progress, assign tasks and to collate all work into a singular space.

One challenge faced was during some group meetings where not fully completed before the meeting. This resulted in less progress in the design process as time was needed to fully complete past elements and work.

We responded by agreeing to have a further look into team member commitments and if someone deemed they would not be able to complete their given section halfway through the sprint, they would leave the completion of their workload to ensure all elements would be completed by the sprint deadline.

Shared conflict here

There was an error that occurred in our code where the SLF4J library suddenly started using a version that was incompatible with the database driver.

At first no one knew why the problem started or how to fix it, so the group responded by not casting any blame, and encouraging all members to work on the error as soon as they were free.

The issue was resolved as a result of combined efforts the following morning.

LL

SLIDE PRESENTED BY:

Team

Do not speak on this slide, live it open on the screen at the end.

State your team name, list team members, and tell us who focused on what parts.

Include a photo next to each name, to help us know who to direct questions to.

Comments from AI on what is missing / what needs to be added after initial draft

◇ 1. Project Description (out of 40 points)

What the rubric expects at the highest level:

- Clear goals, scope, and value of the project.
- All features are clearly described and motivated.
- Technical decisions explained with alternatives considered.
- Technical issues explained with their impact on planning.
- Quality assurance practices discussed with examples.

What your script includes:

- Clear goal of the app: help students study efficiently via gamification.
- Features listed (AI-generated flashcards, exams, quizzes).
- Tools and technologies explained (JavaFX, GitHub, IntelliJ, SQLite, etc.).
- Technical issue example (SLF4J and DB driver incompatibility).
- Some quality assurance: unit testing, GitHub use.
- No discussion of alternatives considered.
- No deeper reflection on quality methods like TDD or refactoring.

Estimated Grade: Above Expectations (34-28 pts)

Likely -32/40 – good coverage, but lacks depth on alternatives and broader QA reflections.

◇ 2. Project Planning, Design, and Teamwork (out of 40 points)

What the rubric expects at the highest level:

- Requirements and user stories listed and prioritised.
- Sprint plan discussed, with response to changes.
- Tools and practices discussed (e.g., GitHub, team meetings).
- Conflicts or challenges reflected on with examples and resolutions.

What your script includes:

- Mentions sprints and planning around checkpoint deadlines.
- Describes team workload balancing and roles.
- Tools: GitHub, OneNote, Discord.
- Conflict examples: incomplete work before meetings, library version issues.
- Response: team stepping in to help, group debugging.

- ✗ Doesn't explicitly mention "requirements" or list user stories.

■ Estimated Grade: Above Expectations (34-28 pts)

Likely -3340 - strong conflict/resolution example, tools well explained, but missing requirements/user stories details.

◇ 3. Delivery (out of 20 points)

Since this is a written script and not the actual live presentation, this section is harder to assess accurately. However, I can rate the **content preparation and clarity**.

Based on the script:

- Structure is clear, logical flow of ideas.
- Technologies, challenges, and teamwork all covered.
- Real-world examples are included.
- ✗ Some grammar/spelling/clarity issues (e.g., "codding", "divided" → should be "divide"; "no one knew..." is misspelled).
- ✗ Needs more polish for professionalism.
- ✗ No mention of slides, visuals, or coordination between speakers in the script.

■ Estimated Grade: Meets Expectations (14-10 pts)

Likely -1320 - solid structure but would benefit from editing, polish, and mention of visuals or delivery coordination.

■ Estimated Total Score:

Category	Score
Project Description	32 / 40
Planning, Design & Teamwork	33 / 40
Delivery	13 / 20
Total	78 / 100

→ Final Grade Estimate: ~78% (Credit or Distinction level)

Suggestions to Improve:

- Add mention of **requirements and user stories** in planning.
- Explain if **other technologies or design ideas** were considered and why they were rejected.
- Improve grammar and clarity in some sentences.
- Mention **visuals** (e.g., architecture diagram, UI screenshots) and how speakers will divide content.

From: <https://d.docs.live.net/0d479abef2944f97/Documentation/Year%2020%20Semester%201/CAH302/Script%20for%20Sec%20Presentation%20-%20Script.docx>

Scrum Minutes

Tuesday, 1 April 2025 11:03 AM

24/03/25 Mon Week 5

Monday, 24 March 2025 11:49 AM

1. Leila and Liv shared possible project ideas
2. Everyone discussed adjustments to them so the project would be something everyone enjoys
 - a. This involved brainstorming these ideas and collating them into a coherent application
 - b. Leila noted these down on the Project Brief page
3. Decided on the final project idea
 - a. Wrote the project requirements together
 - b. Dani and Max came up with additional requirements for features we could add if we have time at the end
4. Set up a shared OneNote
 - a. Lots of troubleshooting - members could access it but only edit it as 'guests'
 - b. Fixed by ensuring the OneNote was created with a QUT Microsoft account, and was shared only with other QUT Microsoft emails
5. Confirmed everyone could access the repository, after it was created by Emily

25/03/25 Tue Week 5

Tuesday, 1 April 2025 11:52 AM

1. Leila and Liv wrote the final, formal project brief
2. Decided on the main characters to be used, after group research
 - a. Named each character together
3. Allocated user stories to be completed
 - a. Liv - High-school students (Competitive, easily overwhelmed, gets bored easily, no time)
 - b. Leila - Uni students (No time, forgetful, appreciates aesthetics, overwhelmed)
 - c. Emily - High-school students (Forgetful, visually impaired, time management, appreciates aesthetics)
 - d. Max - Uni students (Competitive, bored easily, visually impaired, time management)
 - e. Dani - Teachers
4. Decided on work to be done before the next meeting
 - a. Allocated user stories
 - b. Low-fidelity design (Leila and Liv)

31/03/25 Mon Week 6

Tuesday, 1 April 2025 11:17 AM

1. Clarified user story expectations, as everyone interpreted them slightly differently
 - a. Went over completed user stories
 - b. Suggested user stories should be short and simple, rather than large paragraphs, based on lecture content
 - c. Many of us did them as we were taught in high school (lots of detail) but now they are expected to be simple
2. Started writing some combined user stories
 - a. Combined two user stories by Emily and Leila for requirement 1 into one story, for everyone to use as an example
3. Finalised low-fidelity design
 - a. Combined the two individual low-fidelities (by Liv and Leila) into one, going one screen at a time
 - b. For each screen, the team would review both options and discuss the best final design
 - c. Discussed possible fonts and colours - Leila suggested Comfortaa as a font and Emily researched possible colour palettes (see Design tab)
4. Decided on work to be done before the next meeting
 - a. User stories, considering new guidelines
 - b. Medium-fidelity design (Emily and Dani)

01/04/25 Tue Week 6

Tuesday, 1 April 2025 11:03 AM

1. Decided on 'evil' characters
 - a. Everyone researched possible options, and put them on the OneNote for each other to see
 - b. Decided on which ones to use together
2. Decided on 'final boss' character
 - a. Discussed and selected it from the other possible villains
3. Went over java prac about adding to an SQL database
 - a. Discussed how to implement it into our project
4. 3 members completed the in-class practical (Emily, Leila, Max) to then upload to the repository

07/04/25 Mon Week 7

Monday, 7 April 2025 11:04 AM

1. Finalised medium fidelity completed by Emily, and high fidelity by Dani
 - a. Discussed and reviewed them as a team - everyone happy with them
2. Finalised user stories, by going through every one as a group:
 - a. Prioritised it with Moscow
 - b. Added it to GitHub
3. Decided on a sprint plan and recorded on GitHub
 - a. Moved user stories into planned sprints on GitHub
 - b. This division of the items was discussed as a whole team
4. Release plan finished - Liv, Emily and Leila added to the release plan while discussing it as a whole team

08/04/25 Tue Week 7

Tuesday, 8 April 2025 11:09 AM

1. Assigned user stories in product backlog to team members for the next sprint
 - a. Five categories:
 - i. User accounts (sign-in/up) - Liv
 - ii. Quests for different subjects - Max
 - iii. View, edit, delete flashcards - Emily
 - iv. Quizzes - Leila
 - v. Front-end - Dani

Notes From Class Tue Week 7

Tuesday, 8 April 2025 11:39 AM

Continues integration

- Maven and Github Actions

Code base

- Checks that its being built the same way
- Build file can --> requirement for project
 - o Run test
 - o Generates java docs
 - o Deploy to server

Add maven Javadoc plugin (*adds a dependency*)

Publish java docs to github

Modula 10 --> AI will be slowly released after this week

Using Ollama

28/04/25 Mon Week 9

Monday, 28 April 2025 12:00 PM

1. Discussed what was completed over the holiday break
2. Collation and editing of code written individually, so everything could be merged together
 - a. Max helped Liv merge the sign up/sign in code, and solve associated errors
 - b. Leila helped Dani merge the GUI code, and solve associated errors
 - i. Controllers and front-end code written in the same file - had to split up
 - ii. Used an alternative library for the front-end code, different to JavaFX - changed to JavaFX to ensure consistency
 - c. Leila helped to integrate the GUI with the back-end functions
3. Confirmation of video contents
 - a. Discussed what was required for the upcoming video submission as a whole team
4. Allocation of unit tests according to the functions we've already written

29/04/25 Tue Week 9

Tuesday, 29 April 2025 11:15 AM

1. Further integration and testing of GUI lead by Dani
2. Checkpoint
 - a. Feedback
 - i. Add a create flashcards manually option
3. Confirmed preliminary prototype submission roles:
 - a. Leila and Max --> Tidy up and finish the Main UI interface and all other code in sprint 1
 - b. Emily --> Organisation of documentation including report of git commits
 - c. Dani and Liv --> Scripting and filming the video
4. Assigned tasks for sprint 2:
 - a. Character selection for quests & villains in quizzes - Liv
 - b. High scores and achievements - Max
 - c. Add flashcards using AI - Emily
 - d. Quest and flashcard progress bars - Leila
 - e. Front-end - Dani

6/05/25 Tue Week 10

Tuesday, 6 May 2025 11:07 AM

1. Discussed everyone's progress in the current sprint
 - a. Everyone on track so far
2. Leila to also implement CI/CD
3. Max added a new table for achievements in the database
4. Fixed shared database issues
 - a. Errors with Leila's local version where database file became corrupted, because it was included in a merge after changes
 - b. Max helped fix it by deleting the database file and re-running the program to recreate the tables
 - c. Ensured everyone now knows not to commit the database file
5. Fixed issues with recently merged code on main branch
 - a. Liv finalised some of the character selection code

12/05/25 Mon Week 11

Monday, 12 May 2025 1:16 PM

1. Completed merge of code for final submission & checkpoint
 - a. Liv and Leila merged the branch with quiz villains into main
 - i. Fixed merge errors, due to recent changes to the quiz UI
 - ii. Conflicts were mainly in the quiz and quiz-results fxml files, due to changes in the page layouts e.g. introducing StackPanes
 - b. Dani and Leila merged the branch with UI changes into main
 - i. Solved merge errors, caused by the UI changes being across almost all pages and therefore code files
 - ii. Conflicts mainly involved the fxml files
 - c. Emily merged the branch implementing flashcard AI in main
 - i. Solved a conflict in the quest view page due to recent UI changes
 - ii. Involved QuestPageController and quest-page-view.fxml
2. Confirmed completion of all checkpoint content for the whole team
3. Everyone updated the PM tool (GitHub) to mark their items as completed
4. Leila added button styling to the app to finalise the UI before the checkpoint

Questions for tutor

- Do we have to have fully comprehensive java docs?
 - o That means we have to make our repo public?
- Do all people in group have to speak in final presentation
- Does the presentation mirror the submitted video
- Do we and if so when do we get our feedback back on our submitted checkpoint from week 9
 - o On video??
 - o Content??
 - o Overall presentation??

13/05/25 Tue Week 11

Tuesday, 13 May 2025 11:03 AM

1. Assigned script for final presentation - Olivia to write practice script before next Monday for us to use for the practice presentation
 - a. Emily to make the initial PowerPoint for week 12 practice
2. Assigned tasks for the final video submission
 - a. Script - Olivia
 - b. Slides - Emily
 - c. Video demo - Leila
 - d. Video Collation - Dani (batman)
3. Final Checkpoint
 - a. Feedback
 - What was approached and what was the result
 - Summary --> resolve this this and this
 - Otherwise, thought but no results
 - Merging --> source code --> How --> which errors?? (modules, classes, specifications)
 - What information can u get out of the notes
 - Who was the protagonist
 - MORE DETAIL!!!
 - Graded --> how we produced the visibility



How you built the software
Everything even
Upfront
User testing --> others --> feedback --> user acceptance testing

Used ordered list not bullets
Preface of Sarif fonts

Answered Questions

1. Do we have to have fully comprehensive java docs?
 - o That means we have to make our repo public?

Try to without making public, but might just have to

LIVE PRESENTATION:

Quick --> supporting, we may not be looking and just listing

MAX 15 --> talk, set up, grill

2. Do all people in group have to speak in final presentation
 - Yes --> don't have to be equal

3. What questions are to be asked example
 - o How do u code things
 - o Questions about the code specifically?
 - NO - not enough time

Rubric --> for that event --> points if we miss those point we will grill for them

Possibly via prompting that we have answered all the questions

4. Is the demo live / do we need one??

No demo

Software development is what is the presentation

Our app is the Gini pig

5. What is exactly is recorded
 - o Are we filmed??

Slide and audio --> backup and safety net

Script is prompts don't directly read

5. Does the presentation mirror the submitted video
 - o Key differences

Convince that the project works

6. Do we and if so when do we get our feedback back on our submitted checkpoint from week 9
 - o On video??
 - o Content??
 - o Overall presentation??

End of this week we should have grades and feedback --> everything 4 points

1. Object design

2. Code

3. Teamwork

4. Presentation itself

Feedback based on each section of the rubric

Roadmaps to sources

19/05/25 Mon Week 12

Monday, 19 May 2025 11:12 AM

1. Liv, Leila and Emily discussed updates to make to the sprint retrospectives, and possible extra details to provide about what went poorly and what went well
 - a. Finalising past sprint documentation to ensure it is detailed
2. As a group, discussed the final video and task assignments for the final sprint
 - a. Live presentation slide 5, java doc comments, video section based on week 9 - Olivia
 - b. Live presentation slide 1, user testing (1 user), clean up code (remove prints, fix bad formatting), java doc comments, video demo section - Max
 - c. Live presentation slide 4, presentation and video slides, user testing (1 user), java doc comments, video code section - Emily
 - d. Live presentation slide 3, update meeting minutes, java doc comments, video CI/CD section, OneNote export - Leila
 - e. Live presentation slide 2, user testing (1 user), video user testing section, video editing - Danielle
 - f. Presentation and video speaking - EVERYONE

Original notes taken (before formatting):

Max writing notes

Liv, Liela, discussing how retrospectively previous sprints what went poorly. Liv - Had to reallocate/add more tasks for sprint due to time and upcoming tasks.

Group - Going over retrospective for sprint

Liela - further going over TODO for sprints, what was completed.

Group - Discussing video, assigning task for final sprint/finalisation of projects.

Liv - Finishing script for tomorrow practice run.

Emily, liv, Liela , Max - Giving user testing

Dani - Making meeting minutes more detailed

Liv - Discussing what meeting minutes should have in there, more detailed what went wrong, what in the code.

Liela max - Cleaning up code (Formatting, print statements, etc)

Liela - Assigning java doc who is doing what, primary doing there code however for bigger ones where many people worked.

Liv - Creating script next to slides

Emily - creating PowerPoint for presentation

Liv - Creation of the idea, what was our key feature, how are we keeping people engaged.

Liv - Create script

Liela - Discussing finishing off Javadoc's

20/05/2025 Tue Week 12

Tuesday, 20 May 2025 11:02 AM

1. Reviewed week 9 submission feedback and discussed areas for improvement (all teammates)
 - a. Overall visual, sprint lengths, more details in meeting minutes
2. Liv provided the script, for everyone to edit and practice
3. Completed a practice run-through of the presentation to check it's within 5 minutes
4. Everyone made minor adjustments based on team members' speaking styles
5. Everyone reviewed slides made by Emily
6. Ensured everyone had Ollama properly installed - helped Max install it
7. Completed sprint retrospective - Leila and Emily
8. Dani, Leila and Emily made a schedule for the next week, containing deadlines for different tasks throughout the week

Notes on overall videos from tutor:

- Some long winded sections -->
- Don't need / want to know
- Meeting minutes --> helps display team work and how it works
 - Pdf or markdown file
 - Consider formatting
 - Separate file??
- Git log
 - Explain why some have more than others
 - Balance and why so
 - Add more in script about that
 - Why git??
 - Familiarity??
- Conflicts for FV
 - Trivial??
 - Wants to go different ways
 - Need something
 - Nothing works perfectly
- Reflection
 - Same as scums
 - Meetings
 - Dates
- Delivery
 - Engaging
 - Confidant
 - Know what I am talking about
 - Timing --> practice
 - Little body movement is ok
- Slides
 - Stay in formate and slides = probs tick
- Q and A
 - Don't be defensive
 - Here is the answer
 - Be confident
- Dissect the rubric
- EVIDENCE!!!!!!!!!!!!!!

Feedback from assessment:

1. User stories

- a. Looking for wow factor - above and beyond - it meets the criteria, but not exceptional
 - b. More detail within the timeframe - ensuring you hit all the talking points
 - c. How did a specific thing you implemented help the project?
 - d. How did you develop the stories? What technique did you use? e.g. work on as individuals then bring together - did that help you, did that work?
 - e. **That's above and beyond**
 - f. It's ok, but there's room for improvement
 - g. That criteria can't be improved necessarily
2. Git log unbalanced
 - a. Explain why
 3. Breadcrumbs
 - a. Looking for consistency across pages
 - b. The way the program was presented - dull colours is boring
 4. TDD
 - a. List of tests, shows the test code
 - b. At least an indication of github actions for week 13
 - c. Talk about it
 - d. Required to be done - talk about how it was required
 - e. Project requirements: list them - then what options did we have? We had some
 - f. Why did we build what we built? What was part of our thinking?
 - g. On what did we do TDD on? What aspects of our project did we employ TDD? How did you back it up, was it of any use to you? How did it benefit?
 - h. Same with CI/CD - did it find any faults for you? Has it been useful? It was a requirement to do github actions CI/CD - the result of this was... did it provide value? The specific errors/things that happened
 5. Lots of things aren't our choice
 - a. Required to use Java, etc.
 - b. Project specification said we had to use
 6. Slides improvement
 - a. Generally improve
 - b. Simplify background
 - c. Contrast - don't have white bg on white bg

26/05/25 Mon Week 13

Thursday, 29 May 2025 2:43 PM

1. Everyone practiced their live presentation scripts together
 - a. It was consistently around 4min 40sec to 4min 50sec
2. Emily finalised the PowerPoint slides

Agile Documentation

Wednesday, 14 May 2025 3:49 PM

Sprint 1

Monday, 19 May 2025 11:21 AM

End Date:

- Tuesday week 7 (April 8)

Items Assigned:

1. 4 user stories (high-school students - competitive, easily overwhelmed, gets bored easily, no time), low fidelity prototype - Olivia
2. 4 user stories (university students - competitive, bored easily, visually impaired, time management) - Max
3. 4 user stories (high-school students - forgetful, visually impaired, time management, appreciates aesthetics), medium fidelity prototype - Emily
4. 4 user stories (university students - no time, forgetful, appreciates aesthetics, overwhelmed), low fidelity prototype - Leila
5. 4 user stories (teachers), medium fidelity prototype with images and colours - Danielle

Retrospective:

- What went well:
 - The completion and collation of low and medium fidelity prototypes
- What went poorly:
 - The overall structure of the user stories did not compile with the structure provided so all work had to be redone in order to have the correct structure
- What should we do next:
 - Rewrite user stories
 - Disuse and determine possible code functions
 - How will the code be divided
 - Who would be willing to complete each section

Sprint 2

Wednesday, 14 May 2025 3:49 PM

End Date:

- Tuesday week 9 (April 29)

Items Assigned:

1. User accounts (sign-in/up) - Olivia
2. Quests for different subjects - Max
3. View, edit, delete flashcards - Emily
4. Quizzes - Leila
5. Front-end - Danielle

Retrospective:

- What went well:
 - Allocation of tasks - everyone completed allocated tasks by meeting dates, using the PM tool (Git Projects)
- What went poorly:
 - Confusion with git, leading to inconsistent implementation. For example, some wrote all code locally, some used branches, and some committed to main. This lead to many merge/push conflicts which had to be resolved during the meetings.
- What should we do next:
 - Needed to decide on a consistent way to use git, so we wrote up a git procedure for everyone to use whenever they want to make any changes to the code.
 - Also need to remember to write unit tests.

Sprint 3

Wednesday, 14 May 2025 3:49 PM

End Date:

- Tuesday week 11 (May 13)

Items Assigned:

1. Character selection for quests & villains in quizzes - Olivia
2. High scores and achievements - Max
3. Add flashcards using AI - Emily
4. Quest and flashcard progress bars & CI/CD - Leila
5. Front-end - Danielle

Retrospective:

- What went well:
 - The git procedure ensured everyone used git in a consistent manner, so there was less conflicts to resolve during the meeting.
- What went poorly:
 - Tasks were not allocated evenly at first, as items had to be added and reassigned during the sprint. Additionally, not all tasks were completed by the meeting date, so some had to be completed during the following sprint.
- What should we do next:
 - Consider all scheduling factors for each team member before assigning tasks in the next sprints.

Sprint 4

Wednesday, 14 May 2025 3:49 PM

End Date:

- Tuesday week 12 (May 20)

Items Assigned:

1. Script for practice presentation - Olivia
2. Finalise achievements code and UI - Max
3. Edit script for practice presentation - Emily
4. Edit script for practice presentation - Leila
5. Edit script for practice presentation - Danielle

Retrospective:

- What went well:
 - Practiced the script to make sure it flowed and also did a practice run on the final presentation Infront of the tutors which helped us know what to change when it comes to the real presentation
 - Assigned one person to do presentation slides and another to do script was time efficient
- What went poorly:
 - Having one person write the script for everyone meant it didn't suit everyone's peaking styles at first.
- What should we do next:
 - Reevaluate the script for each person, by having all team members work on their own parts of the script.

Sprint 5

Wednesday, 14 May 2025 4:07 PM

Javadocs to finalise:

End Date:

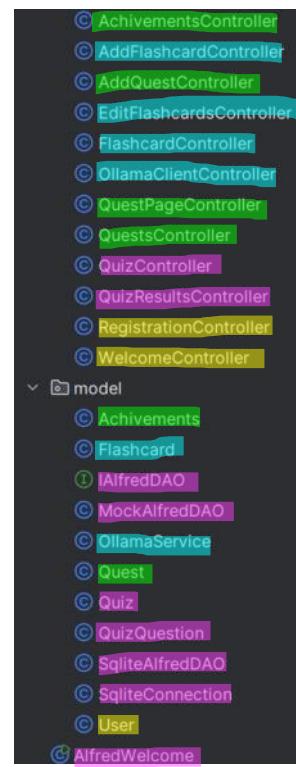
- Friday week 13 (May 30)

Items Assigned:

1. Live presentation slide 5, java doc comments, video section based on week 9 - Olivia
2. Live presentation slide 1, user testing (1 user), clean up code (remove prints, fix bad formatting), java doc comments, video demo section - Max
3. Live presentation slide 4, presentation and video slides, user testing (1 user), java doc comments, video code section - Emily
4. Live presentation slide 3, update meeting minutes, java doc comments, video CI/CD section - Leila
5. Live presentation slide 2, user testing (1 user), video user testing section, video editing - Danielle
6. Presentation and video speaking - EVERYONE

Retrospective:

- What went well:
 - Practice rounds of the speech went well so that we could have it under the 5 minutes for the final presentation
 - Creating a daily schedule for the week leading up to submission
- What went poorly:
 - Check more in depth how much work is required for each task before assigning them to people, as we didn't realise how big/small some of the video parts were going to be
- What should we do next:
 - Finish the final video demo for submission on the 30th of May



TO ASSIGN:

- Demo video
- Documentation
- Presentation slides
- Presentation script
- Java docs and clean up code (remove print statements)
- Update meeting minutes
- User testing

User Feedback

Tuesday 27 May 2025 6:29 am

Page	Positive	Negative	Suggestions
Login	Likes Styling of page	Incorrect login text is green	Make incorrect actions red
Choose Quest Page	Constant styling	Feel's looks empty on first login without any subjects	Send user to a create subject page first if no subjects created yet
Add QuestPage	Likes the look of the characters, Like's simplicity of page	Why to add multilabel end dates for specific exams/assignments not just overall end date	
Quest Page	Likes constant styling and look of progress bar at top, clear what buttons do. Like's the you've travelled X Kms.	Grey background behind enemy, clicking back on all pages bring back to quest page but back on create flash card brings back to quest selection	Fix background colour on characters, wish it was clear where was heading towards
Generation of flash cards	Likes method of creating.	Wish's there was a way to upload a PDF document of coarse content or video from course to get full contexts, making computer slow when creating flash cards freezing stuff	Wishes there was a way to select the amount of flash cards to make, clear loading screen.
Creating own flash cards	Likes being able to add own flash cards	Wants to be able to make more then one without going back to page	Select way to add more then one at a time.
Flash cards	Likes design of flash cards, looping over them not finishing	Confused by mastered flashcards not going up when doing anything on the page, only after being told updates based on the daily quiz	Clear progression
Edit flash cards	Being able to edit all flash cards, and remove them in anyway		Bigger text box for editing
Achievement s	Clear what achievements there are and what has been gained	Simple styling of just text that is either greyed out or solid	Card like structure, and notification system on receiving.

Page	Positive	Negative	Suggestions
Login	Cute and appealing front page	Register might need to standout more	Change colour of register here text
Choose Quest Page	Same as login	Confused where to click on subjects, not separated	Have clear separation lines between subjects, more like buttons
Add QuestPage	Loves the designs of all the characters and their names, makes it more personable		
Quest Page	Intrigued by the fantasy feel of the app, loves the map like navigation	Colour is a bit weird around the boss, out of place	Blend into background (boss)
Generation of flash cards	Enjoys the efficiency of generation	Takes a bit long to generate	
Creating own flash cards	Likes that your able to add your own as well as generate		
Flash cards	Clean and easy navigate able design		
Edit flash cards	Likes that if there is a mistake in ai generation or creating your own you are able to make changes to current or delete irrelevant ones	Pop ups aren't as aesthetically pleasing as the app	
Achievements	Likes that you can keep track of and work towards achievements	Could be more on theme elements	Make the page more aesthetically pleasing through characters and more exciting flashy items for achievements
Overall	Enjoys the gamification of the app and makes study more of a fun experience		

Monday, 24 March 2025 12:08 PM

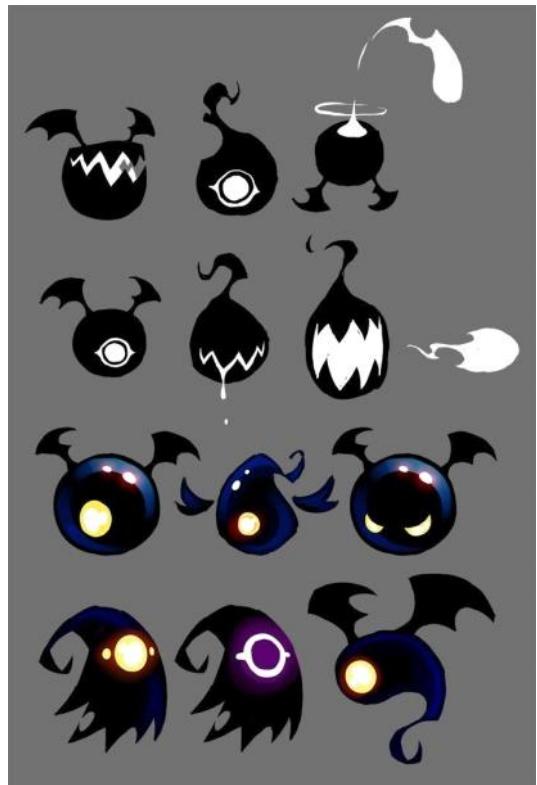
Characters

Monday, 24 March 2025 12:04 PM

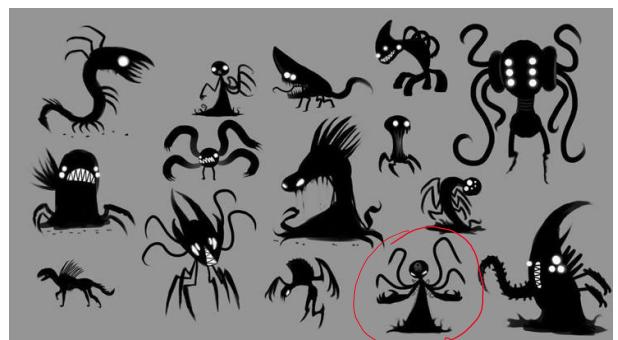


Bad Guys

Monday, 31 March 2025 2:34 PM



could use this
for when they're
dead/defeated?

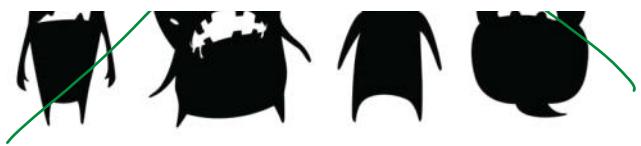


← maybe use one
of these for the
final boss?



w





Edited Characters

Tuesday, 6 May 2025 11:14 AM

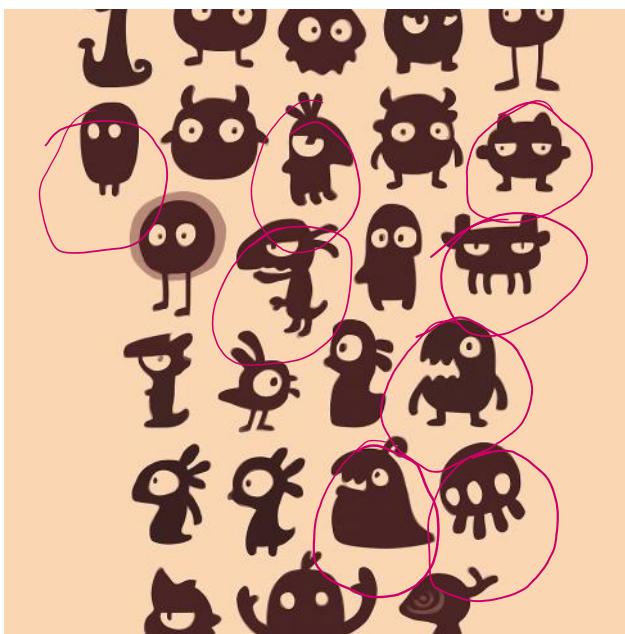


Take capi from 2

Bear from 2

Oscar from 2





Edited Villain

Wednesday, 14 May 2025 10:14 AM



UI Design

Monday, 31 March 2025 2:34 PM

Low Fidelity - Liv

Tuesday, 25 March 2025 11:40 AM

Login / welcome

Welcome To:
ALFRED AI

Ready to Continue the Adventure

Username:

Password:

Submit

Not an adventurer yet? Register

Main Page

Alfred AI



Welcome back, how will you continue your adventure?

Subject 1 Subject 2 Subject 3

Subject 4 **Add new subject**



Subject name

Subject home page

Are you ready to continue our Subject Name adventure?

40 XP

How Do You Wish to Continue our Adventure?

Prep for your adventure by getting supplies Depart on an adventure

flashcards

exam multiple choice 

Chest or monster image



Question 10

10. ~~~~~

a. ~~~~~

b. ~~~~~

c. ~~~~~

Flash cards

lets prep for your next adventure

40 XP

Question	Answer

Add new flash card

Exam long response



Question 10

10. ~~~~~

Submit

Low Fidelity - Leila

Sunday, 30 March 2025 3:13 PM

Welcome to
Alfred AI!

Ready to continue your adventure?

Username

Password

Not an adventurer yet? [Register here](#)

Start your adventure!

Username

Password

Confirm Password

A curved arrow points from the "Register here" link in the first box down to the "Start your adventure!" box.

Welcome!

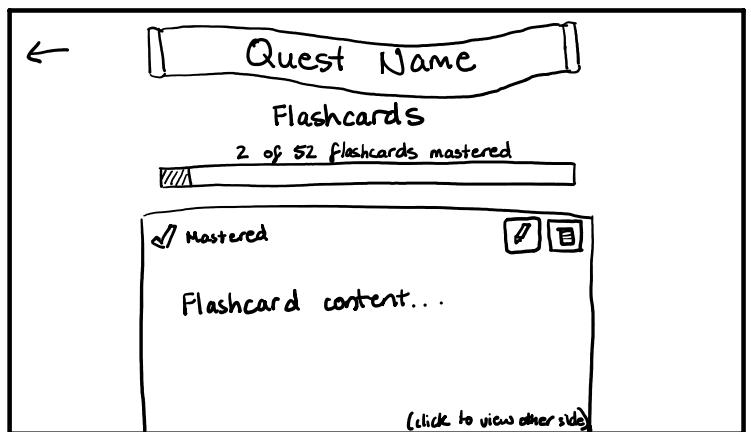
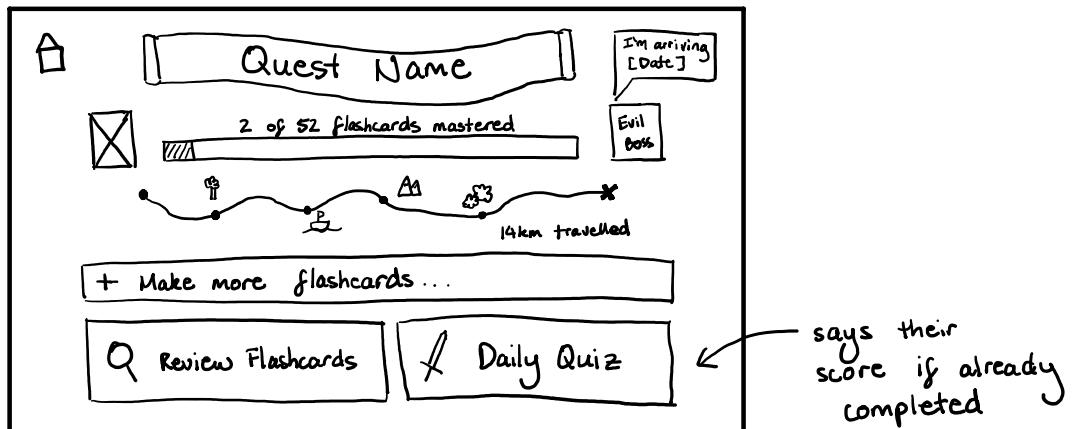
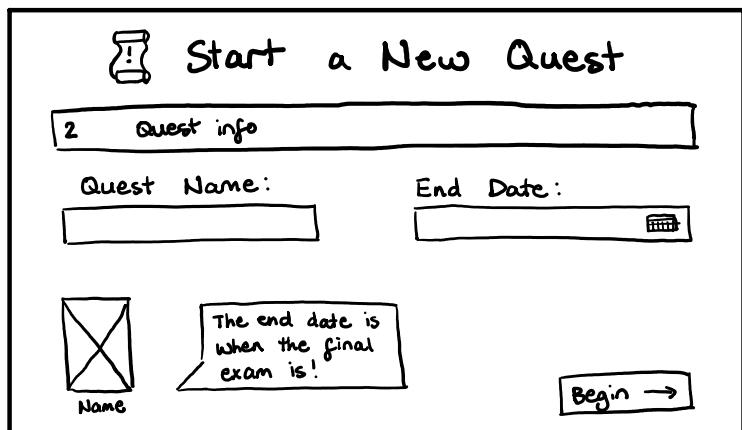
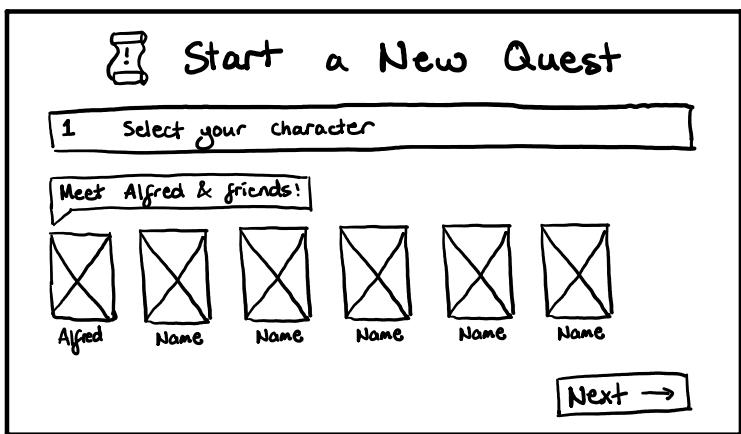
+ New quest

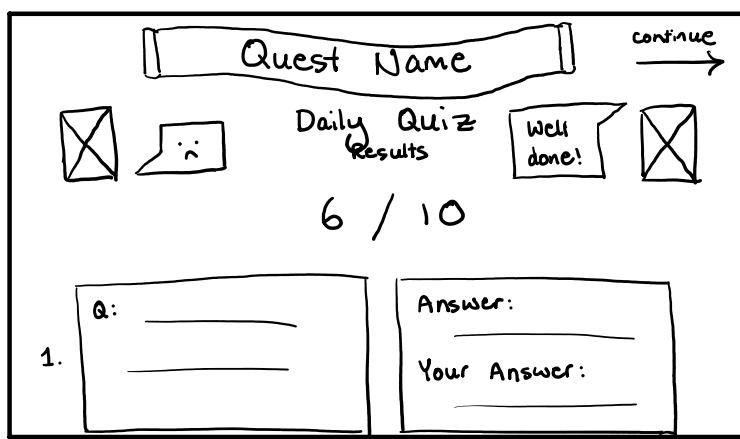
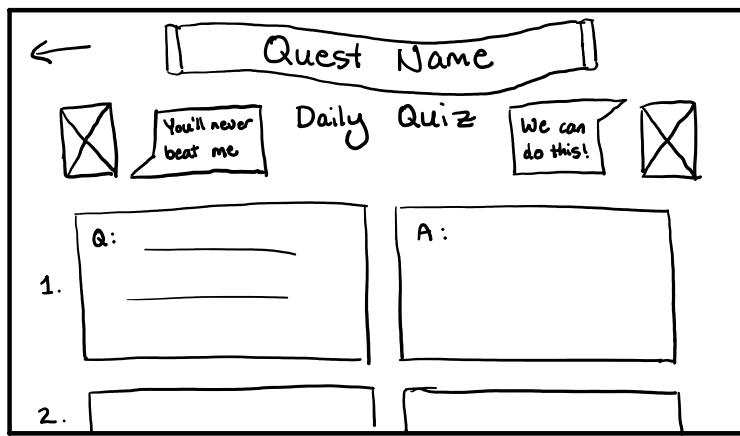
Subject 1

Subject 2

← TO DO:
show high scores

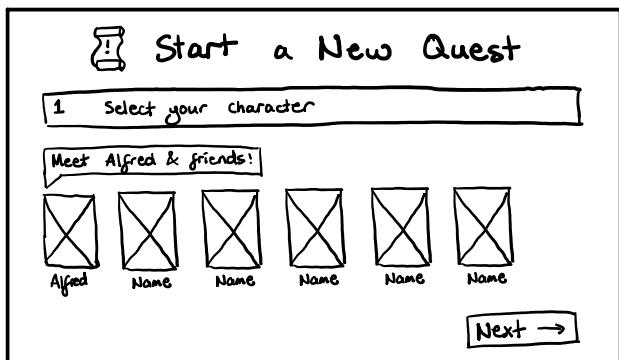
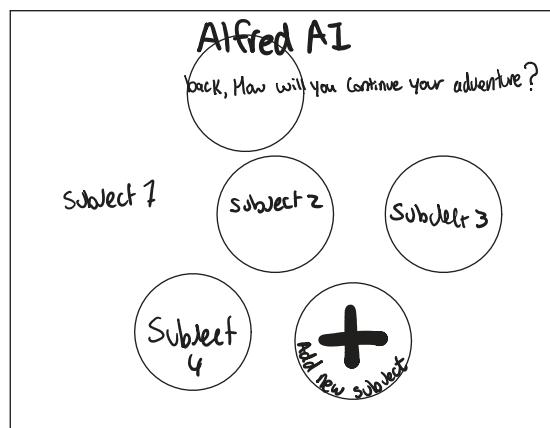
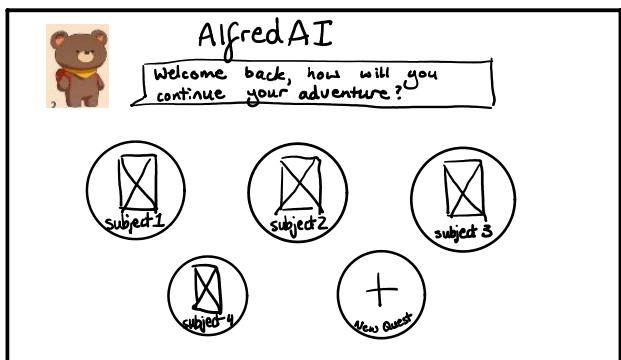
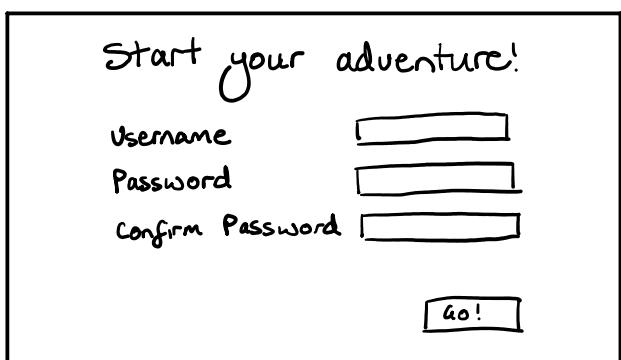
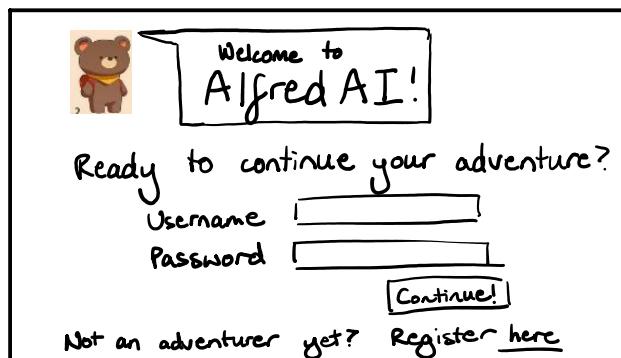
character icons





Final Low Fidelity

Monday, 31 March 2025 10:49 AM



Start a New Quest

2 Quest info

Quest Name: _____ End Date: _____

Name 

The end date is when the final exam is!

Begin →

Quest Name



I'm arriving [Date]

Evil boss

+ Make more flashcards...

Review Flashcards Daily Quiz

M 6/10



Quest Name

Flashcards

2 of 52 flashcards mastered

Mastered

Flashcard content...

(click to view other side)

(scroll down to view more)

Quest Name

Edit Flashcards

Let's prep for your next battle!

Question	Answer
_____	_____ / 目
_____	_____ / 目

lets prep for your next adventure



Question	Answer

④ Add new flash card

Quest Name

Daily Quiz



Q: _____	A: _____
1. _____	_____
2. _____	_____



Quest Name

Well done!

Daily Quiz Results

6 / 10

Q: _____

Answer: _____
Your Answer: _____

continue

Medium Fidelity - Emily

Monday, 31 March 2025 9:55 PM



Welcome to
Alfred AI !

Ready to continue your adventure?

Username :

Password :

Continue !

Not an adventurer yet? Register [here](#)

Start your Adventure !

Username :

Password :

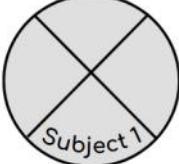
Confirm Password :

Go !



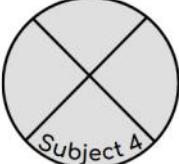
Alfred AI

Welcome back! How will you
continue your adventure ?

 Subject 1

 Subject 2

 Subject 3

 Subject 4

 + New Quest



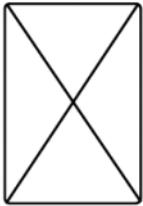
Start a new Quest!

1. Select your character

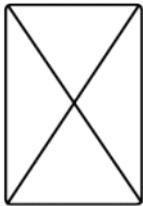
Meet Alfred & friends !



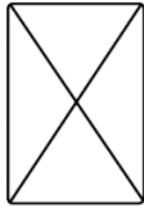
Alfred



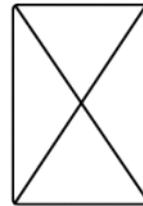
Name



Name



Name



Name

Next →



Start a new Quest!

2. Quest Information

Quest Name :

End Date :

Name

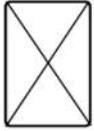
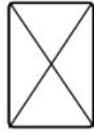
The end date
is when the
final exam is !

Begin →



Quest Name

I'm
Arriving
[Date]



+ Make more flashcards ...



Review Flashcards



Daily Quiz



[High Score]

◀

Quest Name

Flashcards

2 of 52 Flashcards mastered

✓ Mastered

[Flash card content]

(Click to view answer)

▶

◀



Quest Name

Edit Flashcards

Let's prep for your next battle !

Question	Answer

▶

◀

Quest Name

Daily Quiz

1.

Q: _____

A: _____

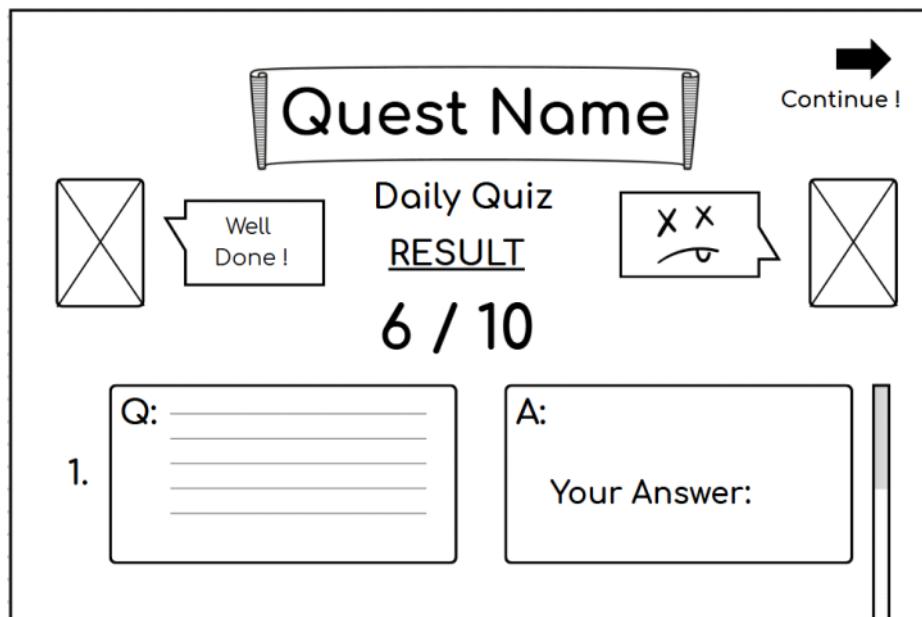
2.

Q: _____

A: _____

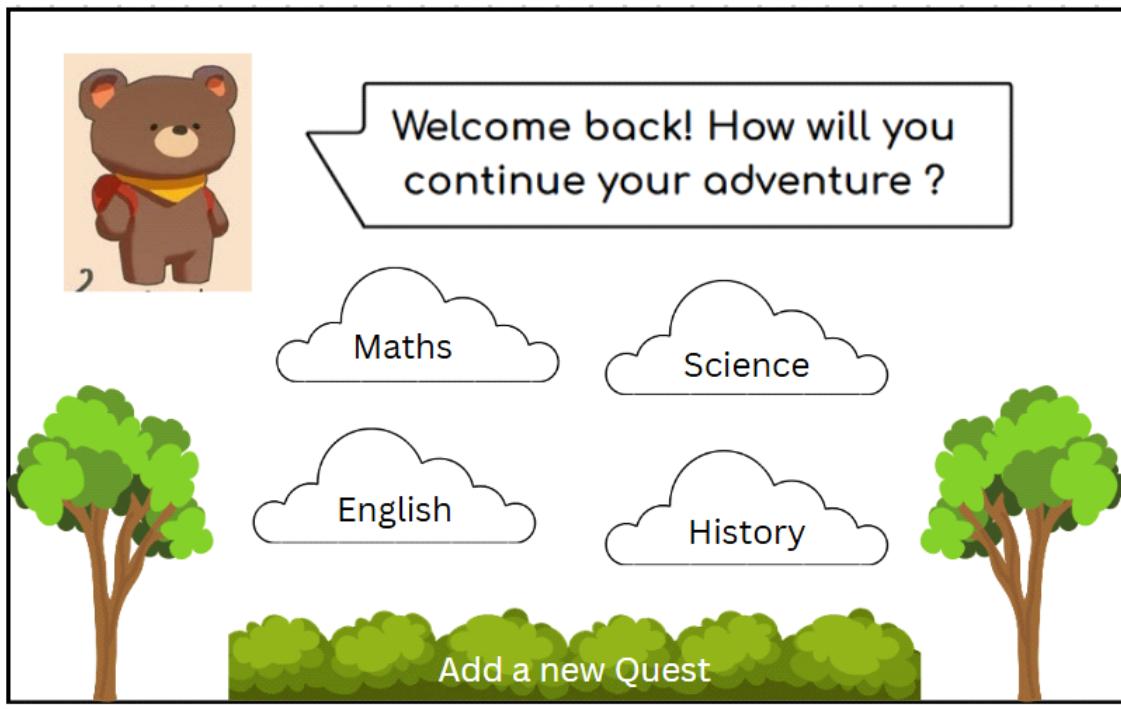
✖

▶



Medium/High Fidelity - Dani

Tuesday, 1 April 2025 11:04 AM



A character selection screen. On the left is a scroll icon with an exclamation mark. The main title is "Start a new Quest!". Below it is a step 1 box labeled "1. Select your character". The sub-instruction "Meet Alfred & friends!" is followed by five character cards: Alfred (brown bear), Penny (white dog), Molly (orange cat), Harry (white dog wearing a hat), and Capi (orange dog). Each character has a small number above them (2, 6, 4, 3, 8). At the bottom right is a "Next →" button.



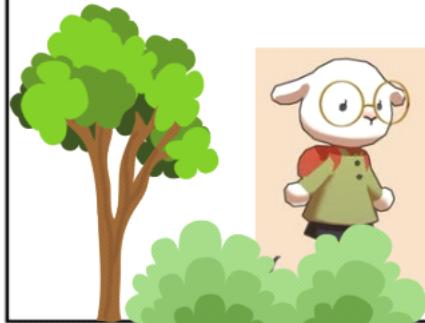
Start a new Quest!



2. Quest Information

Quest Name :

End Date :



The end date
is when the
final exam is!



Begin →



Quest Name

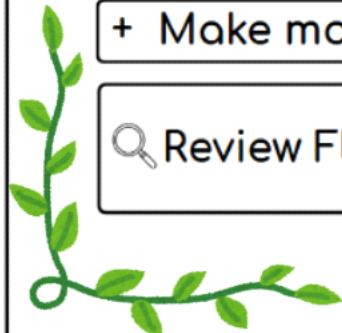
I'm
Arriving
[Date]



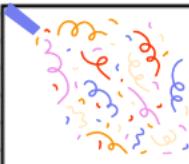
+ Make more flashcards ...

Review Flashcards

Daily Quiz



Crown [High Score]



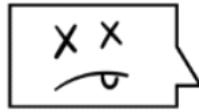
Quest Name

Continue! 



Well Done!

Daily Quiz
RESULT



6 / 10

Q: _____
1. _____

A:
Your Answer:

Font Ideas and Aesthetics

Monday, 31 March 2025 9:36 AM



Database Design Ideas

Friday, 11 April 2025 10:15 AM

Users (**ID**, username, password)

UserQuests(**userID**, **questID**)

Quests(**ID**, character, name, endDate, distanceTravelled, lastQuizScore, lastQuizDate, highestQuizScore)

QuestFlashcards(**questID**, **flashcardID**)

Flashcards(**ID**, question, answer, mastered)

+achievements to be added

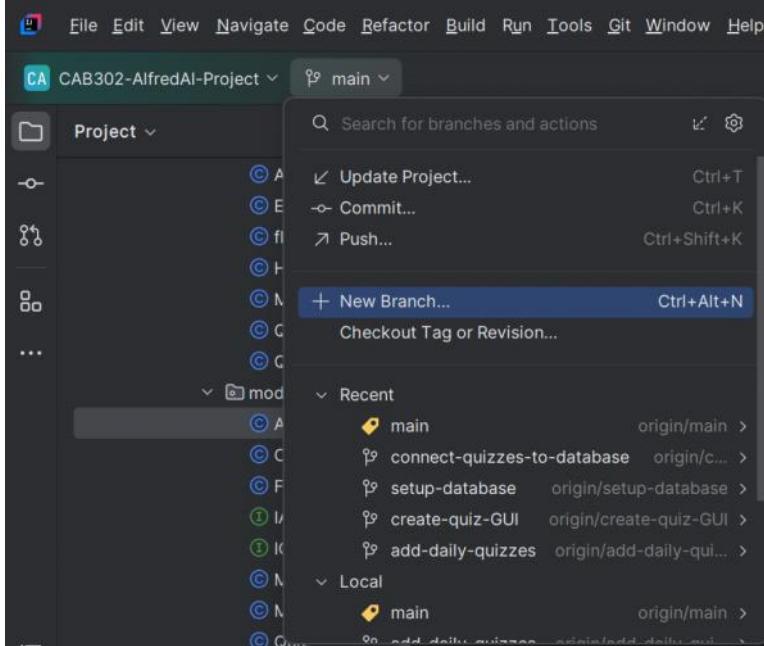
GitHub Procedure

Monday, 28 April 2025 11:30 AM

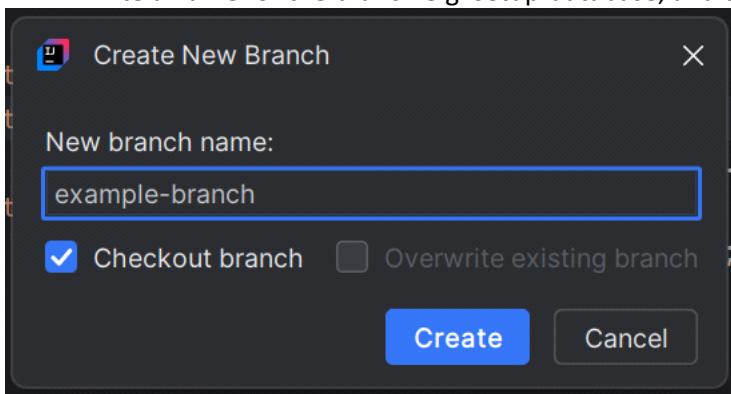
If you want to make a change to the repository:

Make a new branch

1. Click the **+ New Branch** button below.



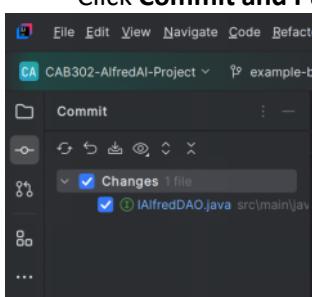
2. Write a name for the branch e.g. setup-database, and click **Create**.

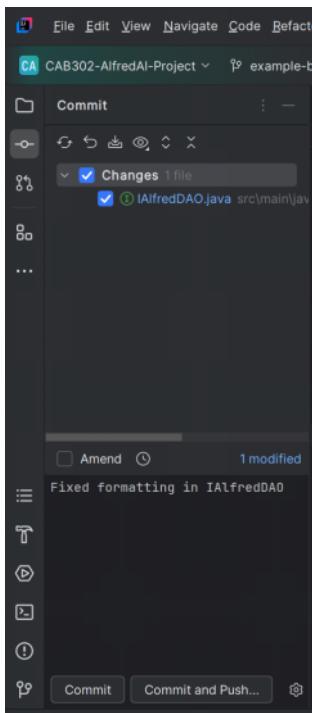


After making any small change, commit and push your change

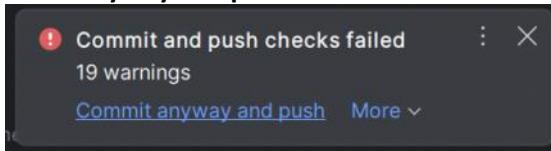
- Committing saves the change to your local copy of the branch.
- Pushing saves the change to your branch on the cloud, so if anyone else views your branch they can also see the change.

3. Select all the files with your changes you want to save (probably all of them). Click **Commit and Push**.

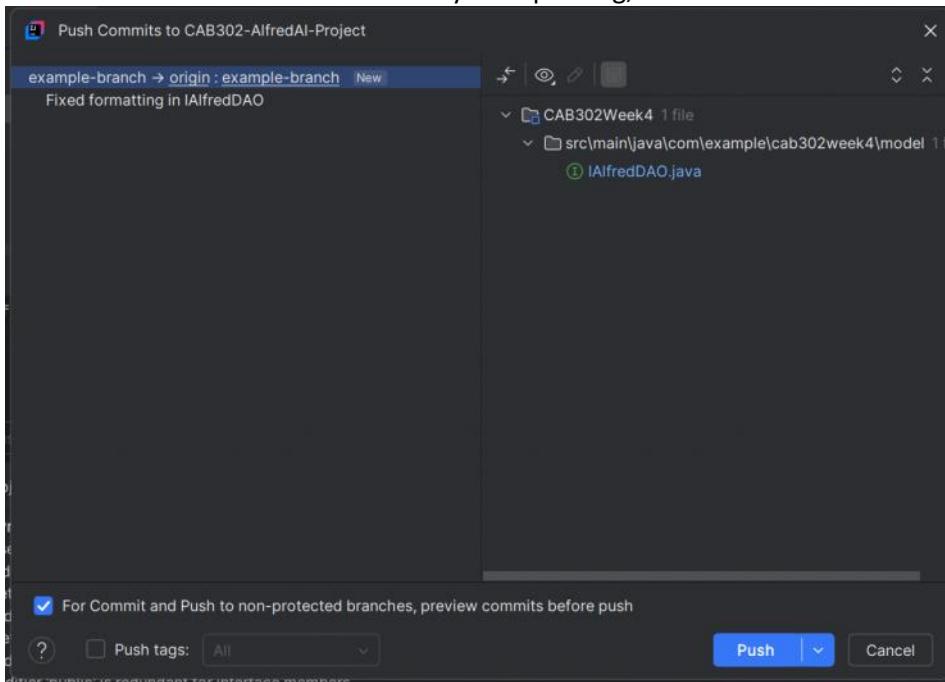




4. If the commit and push fails, you can click **More** and **review code analysis** to see warnings and errors. If the warnings don't matter, select **Commit anyway and push**.



5. You can look over the commits you're pushing, and click **Push**.



When you're finished with your branch and ready to merge with the main branch...

6. Push your final change as above, then go to the repository on GitHub. There should be a notification that asks if you want to create a pull request. Click **Compare & pull request**.

The screenshot shows a GitHub repository page for 'CAB302-AlfredAI-Project'. A pull request from 'example-branch' to 'main' is displayed. The pull request has 52 commits. The commit list includes changes in 'idea', 'mvn/wrapper', 'lib', and 'src/main' files. One commit removed a duplicate declaration in 'MainController'. Another commit updated a file named '.gitignore'. The pull request is currently mergeable.

- Check the pull request is from your branch into main
(It says main <- example-branch)

The screenshot shows the 'Open a pull request' interface on GitHub. The 'base' dropdown is set to 'main' and the 'compare' dropdown is set to 'example-branch'. A green checkmark icon indicates that the branches are 'Able to merge'. Below this, there are fields for 'Add a title' (containing 'Fixed formatting in IAlfredDAO') and 'Add a description' (with a rich text editor and a placeholder 'Add your description here...').

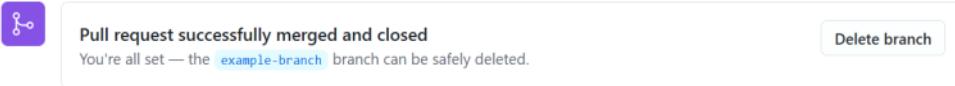
- Write a title, scroll down and click **Create pull request**.

- Check there's no conflicts with the base branch.
 - If there is, try to resolve them? If you can't maybe ask for help.
 - If there is no conflicts, click **Merge pull request**.

The screenshot shows a merged pull request on GitHub. A message from 'Imraad' states: 'Fixed formatting in IAlfredDAO'. The merge summary indicates 1 commit was merged from 'example-branch' into 'main'. The commit message was 'Fixed formatting in IAlfredDAO'. A note at the bottom says 'No conflicts with base branch' and provides a 'Merge pull request' button.

- Write a commit message (can just use the auto-generated one) and click **Confirm merge**.

11. It should then show this:



- You can now safely delete the branch. Click **Delete branch**.

To Do Wk 13

Tuesday, 20 May 2025 12:53 PM

Date Due	Friday Wk 12	Monday MEETING Wk 13	Tuesday Wk 13	Tuesday NIGHT Wk 13	Thursday Wk 13	Friday Wk 13
Emily	• User testing 1 user	• Java doc comments • Finish setting up GH pages (7.3.2 Activity 5)	• Live presentation slide script • Live presentation practice • Live presentation PowerPoint	• Video script for code section	• Video PowerPoint • Video recording	
Max	• User testing 1 user	• Java doc comments • Clean up code (remove prints, fix bad formatting)	• Live presentation slide script • Live presentation practice	• Video script for demo section	• Video recording	
Leila		• Java doc comments • Check GH pages all good	• Live presentation slide script • Live presentation practice	• Update meeting minutes • Video script for CI/CD section • Video script for code section	• Video recording	
Dani	• User testing 1 user		• Live presentation slide script • Live presentation practice	• Video script for user testing section	• Video recording	• Video editing
Liv		• Java doc comments	• Live presentation slide script • Live presentation practice	• Video script for section based on week 9	• Video recording	

Highlighting:

- Live presentation
- Video
- Finished

(For java docs see Sprint 5 page)