

Study Buddy (Report)

*Online Study Tracker*

Table of Contents

[**Background**](#_heading=h.tyjcwt) **3**

[**Planning and Research**](#_heading=h.3dy6vkm) **4**

[**Prototyping and Iteration**](#_heading=h.1t3h5sf) **5**

[**Design**](#_heading=h.3j3ykvj09wh2) **8**

[Front-End](#_heading=h.2s8eyo1) 8

[Back-End](#_heading=h.17dp8vu) 9

[**System Development**](#_heading=h.3rdcrjn) **10**

[First Sprint (Register & Log-In Users)](#_heading=h.26in1rg) 10

[Front-End](#_heading=h.lnxbz9) 10

[Back-End](#_heading=h.35nkun2) 10

[Testing](#_heading=h.1ksv4uv) 10

[Second Sprint (Overview & Add Module)](#_heading=h.44sinio) 10

[Front-End](#_heading=h.2jxsxqh) 10

[Back-End](#_heading=h.z337ya) 10

[Testing](#_heading=h.3j2qqm3) 10

[**Analysis**](#_heading=h.1y810tw) **13**

[**Evaluation**](#_heading=h.4i7ojhp) **15**

[Process](#_heading=h.2xcytpi) 15

[**Product**](#_heading=h.5iz04j7y6re) **16**

[**End User Product Evaluation**](#_heading=h.q0wo94do2ks9) **16**

[**Internal User Product evaluation**](#_heading=h.jxyi6bz46izq) **17**

[Front-End](#_heading=h.3whwml4) 18

[Back-End](#_heading=h.2bn6wsx) 18

[Testing](#_heading=h.qsh70q) 18

[**Conclusion / Summary**](#_heading=h.3as4poj) **21**

[**Outcomes**](#_heading=h.o36pdoztc0m) **21**

[**Development Practice**](#_heading=h.ie5w4e3z0u6k) **21**

[**Future Plans**](#_heading=h.ueqyzti5f3fp) **22**

[**Appendix - Marking Rubric (To be deleted)**](#_heading=h.1pxezwc) **23**

[Technical (30)](#_heading=h.49x2ik5) 23

[Testing (10)](#_heading=h.2p2csry) 23

[Report (30)](#_heading=h.147n2zr) 23

[**Appendix II - Brainstorming & Notes (To be deleted)**](#_heading=h.3o7alnk) **25**

[**Appendix III - Backlog & Sprint Planning (WIP)**](#_heading=h.23ckvvd) **26**

# Background

Andreas

// ONLY FOR INSPIRATION AND GUIDANCE. WILL BE DELETED.

Here you should detail your agenda including your aims, literature and sources of information and the scope of your project. This should set the scene for the work that follows including ideation steps and a brief summary of your group and process.

---

6.104 Teamwork activity and deliverable: defining principles

Task 1

Try to define your principles as a team. Who are you and what sort of things do you believe in?

- Can you rank your principles in accordance with how important they are?

- Are there other stakeholders’ perspectives that need to be considered?

- Do you see yourselves as an agile team or is a more rigid structure appropriate for your work?

The outcomes of this activity are:

- a list of top five principles ranked in order

- a list of stakeholders' perspectives to consider.

---

Make some notes on how you can leverage collaborative feedback to improve your designs and converge on your goals in a systematic, iterative way. Try to identify 2-3 risks of the project and lessons learned in the first half of the course. How can you avoid making the same mistakes going forward?

# Planning and Research

Andreas

// ONLY FOR INSPIRATION AND GUIDANCE. WILL BE DELETED.

This section should focus on evidencing the research that you have undertaken in order to substantiate the project. This could include a literature review, some visual evidence of milestones and process (e.g. Gantt charts) and any evidence of planning and iteration. It should be clear to the reader how and why you have settled on certain approaches for your project and how resources and time are allocated in relation to the functionality of your system. There should be a detailed breakdown of your plan in terms of a working process and set of expectations for deliverables (e.g. dates, times, functionality etc.)

---

6.202 Teamwork activity: project tasks distribution

Task 1

It is time as a team to start building the thing that we proposed! At this stage, we have:

- a proposal that informs us about research, some design iteration and some concepts or ideas that we would like to bring to fruition

- standard ways of working, defined through various activities that should be actionable and accountable

- a range of possible ways to implement our design solutions.

Now we need to set the groundwork for the body of work that comes. Discuss as a group and reflect on your plans so far (e.g. Gantt charts, project proposal, timelines and list of requirements). You should start to put some structure in place for how you are going to realise your goals.

Task 2

Now that you have your principles in place, it is time to think about how you will complete the work as a team.

- Who will be tasked with what?

- Are there any dependencies in terms of features/resources?

- How can you ensure a fair breakdown of work?

- What strategies will you use to ensure conformity with the agreed upon principles?

This work will form the basis of your sprints to come. You will be defining working timescales and deciding when is the right time to reflect on your practices. Take some time on this! It is important!

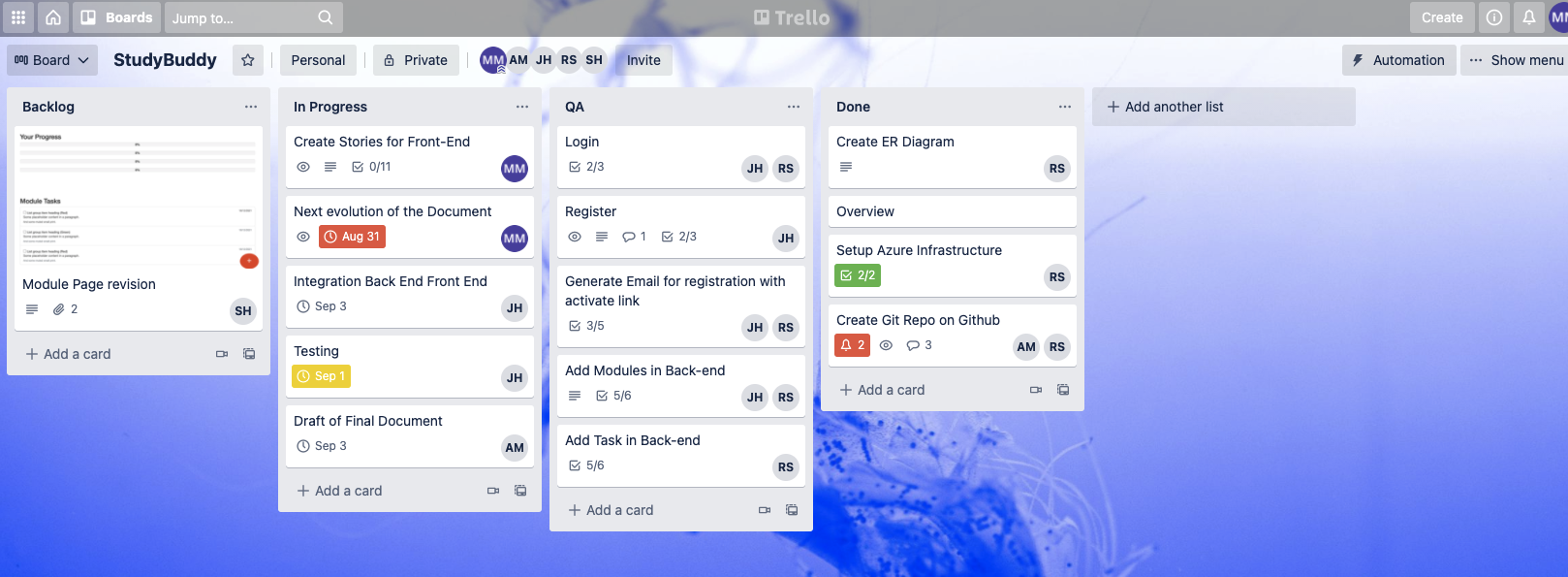
# Prototyping and Iteration

Max

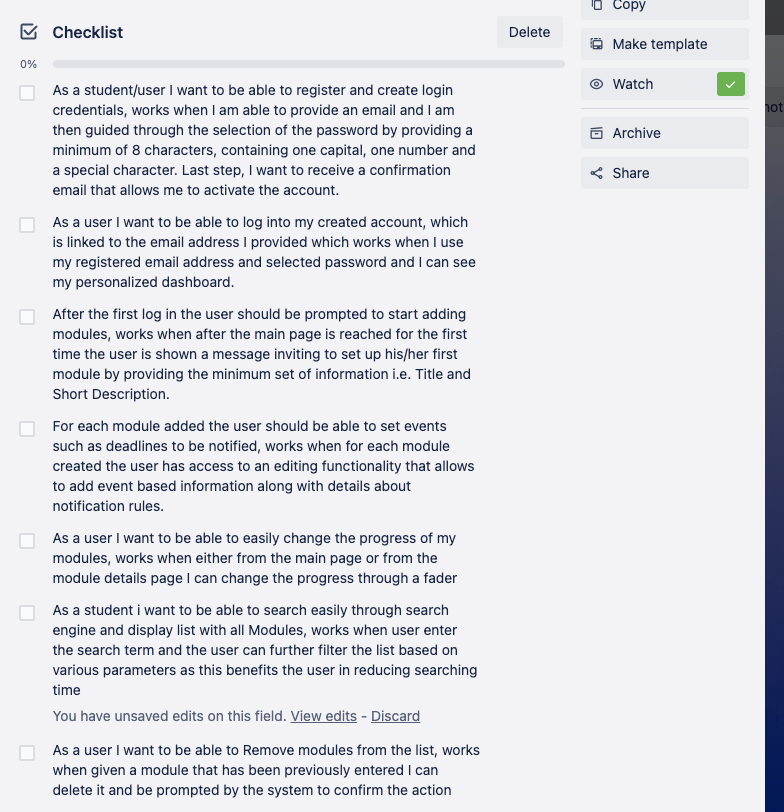
// ONLY FOR INSPIRATION AND GUIDANCE. WILL BE DELETED.

Should show evidence of iteration from low, to medium to high fidelity prototyping. There should be clear rhetoric, justification for design decisions and a process of systematic evaluation that results in clear improvements to the system design over time and iteration. Equally, functional prototypes should evidence the development of code over time including testing of input/output. Functionality should be justified through an analytical approach to development e.g. “We added this feature because...”

The prototype was built through two scrum iterations, below a screenshot of the kanban board taken as one of the first iteration was almost completed:



The activities were organized as user stories configured as tasks for each card as shown in the screenshot below.



In order to align the developments with the scope drafted for Midterm the scope features where detailed and used and used as guidelines as shown in the screenshot below.

# 

Github was used to version control the code

# 

# Design

## Front-End

Swalin

**Front-End Design Phase**

The design phase will utilize the user needs analysis in order to make the most effective product possible. This phase will focus on the specificity of the Study Buddy app (Study Tracking) and how it can be made in the most user-friendly way possible, as well as optimizing its efficiency. It is crucial in this phase that all targeted users (Middle Age Students) can use this app efficiently.

**Front-End Design Purpose**

Our Design major goal is to keep track of the necessary information to properly describe the architecture of our system in order to provide direction and guidance to the development team on the architecture of the system to be developed.

Our Front-end Designs are incrementally and iteratively produced during the *SDLC* (system development life cycle), based on the particular and targeted user feedback.

**Front-End Design Languages**

After several meetings and discussions, we made the decision to employ these front-end languages.

1. **HTML** will be used for the structure of the page and will consist of a series of different elements to show the intended structure of the page. It also provides a number of benefits that will support our project.

a. The browser would be *unsure and unable* to display the content correctly, without HTML elements and structure.

b. It’s Ability to Integrate with other languages.

c. Simple to use

2. **CSS** will describe how the HTML elements on the site are displayed and presented onto the screen. It also provides a number of benefits that will assist our project.

a. Using CSS can *save and reduce the amount of work*. This can be done by controlling the layout of multiple web pages all at one time.

b. Multiple Browser Support

c. Speed

3. **JavaScript**, one of the most important and popular of front end development languages due to:-

a. Its ability to *create dynamic and responsive web pages*. Not only are interactive web pages fantastic for user experience, but this useful addition can also play dividends when it comes to paid advertising.

b. Its capability to give you Extended Functionality.

4. **JQuery** is a JavaScript library that allows us to write less code and create sleeker sites. With a unique ability to simplify programming, it’s more favourable due to:-

a. Its ease and user-friendliness.

b. Cross-browser compatibility.

c. Fast page Speed.

5. **JSON** is JavaScript Object Notation. It will be used for serializing and transmitting structured data over a network connection. It’s primarily to transmit data between a server and web application. It is a much-more compact way of transmitting sets of data across network connections as compared to XML

## Back-End

Robert

// ONLY FOR INSPIRATION AND GUIDANCE. WILL BE DELETED.

This section should focus on how you have come about your design specifications through prototyping and evaluation. You should (re)detail the specifications of your system at this point with a clear view about what it is you are building and the approaches that you intend to take. It should be clear what you are intending to build and how you intend to go about producing such a build in the time frame given.

# System Development

Intro to be added here by Andreas

## First Sprint (Register & Log-In Users)

Intro to be added here by Andreas

### Front-End

Swalin

### Back-End

Robert

### Testing

Jorge

## Second Sprint (Overview & Add Module)

Intro to be added here by Andreas

### Front-End

Swalin

### Back-End

Robert

### Testing

Jorge

// ONLY FOR INSPIRATION AND GUIDANCE. WILL BE DELETED.

There should be a clear and concise description of the development of code, assets and resources that make up the core components of the system. This should include any steps or stages in your development process and testing. You can make reference to your central git repository here, though you might want to include snippets of code and/or processes to better describe how outcomes were achieved. You should also describe the processes that you engaged in e.g. test-driven development, agile techniques and how they enabled the development and iterative, incremental changes that contributed towards the success/failure of your projects.

---

6.203 Teamwork activity: development time

Task 1

As a team, use this time to start to work on the core features of your software application. You might want to institute this as a sprint (e.g. one week, two weeks) or to define more clearly steps and stages of development using a different iterative strategy.

Make sure you think about communication and collaboration including things like ownership of tasks and prioritisation of deliverables. Try to document your process as you go.

---

7.103 Teamwork activity and deliverable: strategies for testing the functional aspects of software

Task 1

In your team, discuss ideas around strategies for testing the functional aspects of your software. When should you test? How often? What sorts of tests happen at what stages?

The outcomes of this activity are:

- a methodology and frequency of testing the functional aspects of your software. You should decide on this together.

- a list of specific tests to take place in the defined stages. These will likely be unit tests, but you can use other types of technical tests if you prefer.

- what are the implications for a failed test? Should someone take ownership of tracking issues?

You might also want to decide if test-driven development is right for your context. Factors such as the framework you utilise might make this easier/harder to implement - so make choices that work with your given workflow.

---

7.203 Teamwork activity and deliverable: designing and running tests

Task 1

In your team, design and run some tests on your software. The outcomes of this activity are:

- keeping a record of the tests and the feedback you have received from testing

- a list of aspects that work well i.e. parts of the system that meet certain criteria

- a list of aspects that might need tweaking or changing based on the feedback from your tests.

Remember, not all tests are created equal! Your tests may be about getting an expected input, reaching a state in a given threshold or timeframe, or may be to do with how components of your system interact and pass data around. This all depends on the application that you are building but should be well justified.

---

8.02 Teamwork activity: project progress

Now might be a good time to think about the progress you have made thus far. It is important that we look back on what we have done with a view to progressing and completing our incomplete tasks. As projects move along and grow in scope, it is important that you partake in these reflective exercises.

How far along are you in your project?

What elements are missing?

How will you divide your time and resources to ensure that you deliver on at least your minimum viable product?

If you are struggling to complete things in a timely manner, you might want to reflect on things like the allocation of work, the scope of the project and any contingency that needs to be put in place.

# Analysis

Max

// ONLY FOR INSPIRATION AND GUIDANCE. WILL BE DELETED.

You should produce a comprehensive analysis of your solution(s) relating to a wide range of resources including literature.

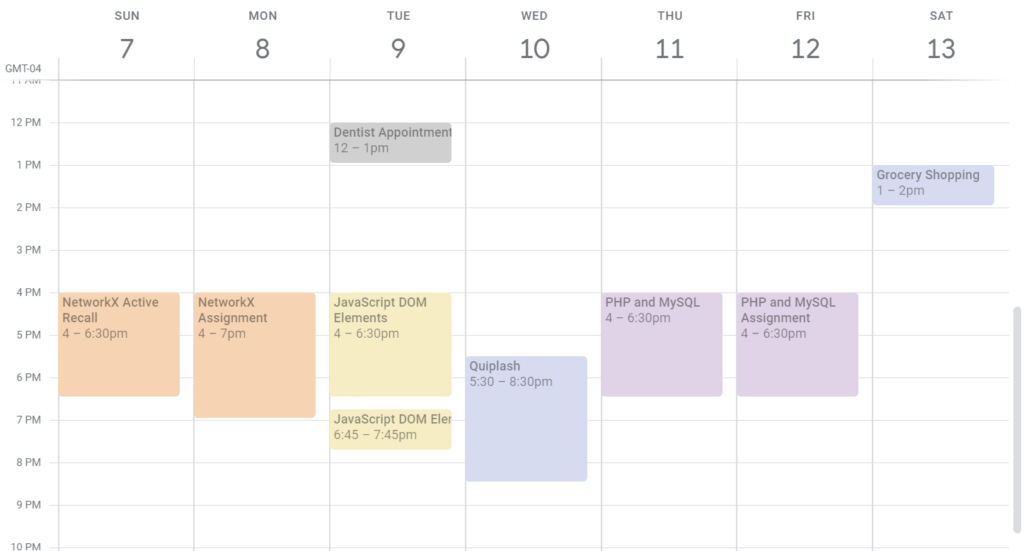
The aim of the application being prototyped is to improve the ability of students to track their progress and has been specifically designed for students of all ages to gain control over their studies in a more efficient way than the traditional general purpose time management and productivity software.

In the following example modules.org allows to create modules and create a calendar for the lessons

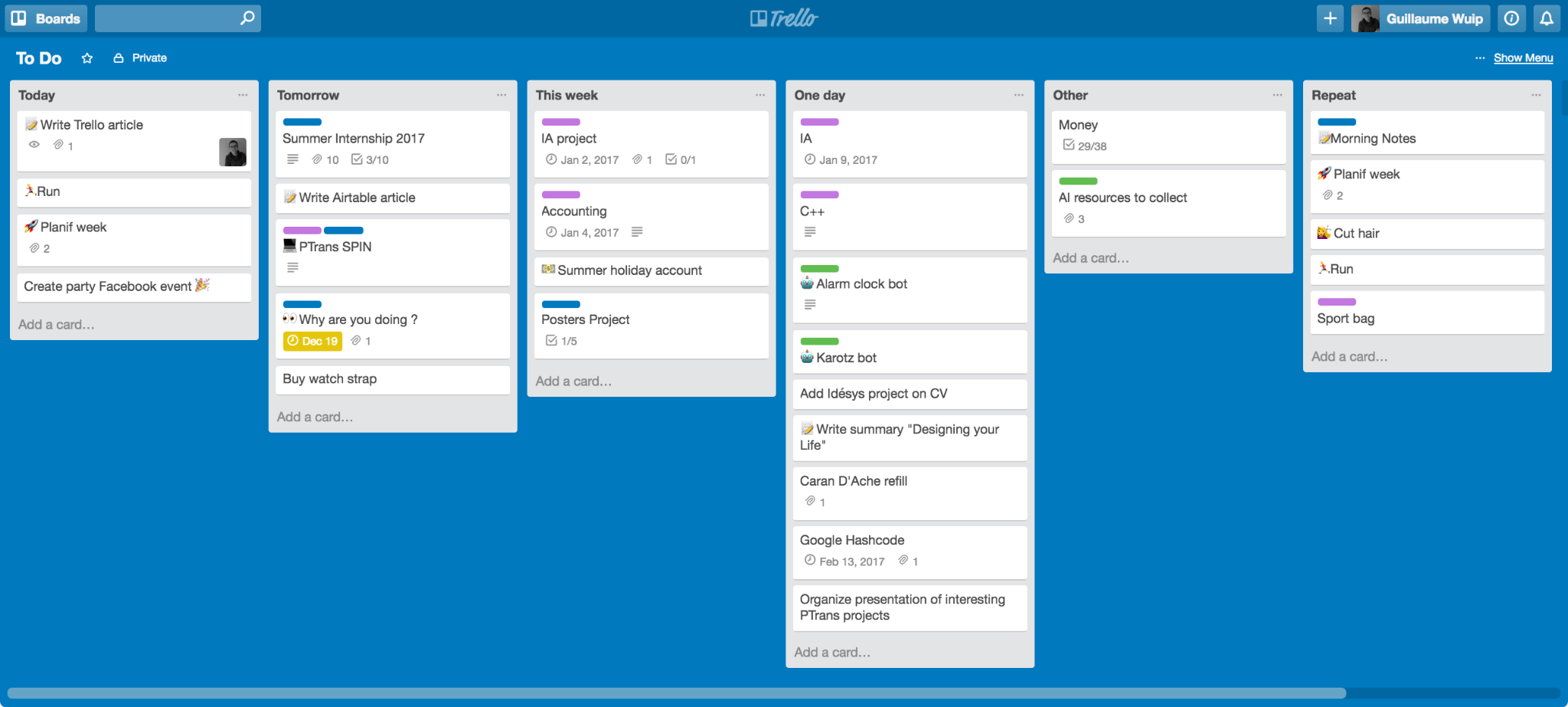


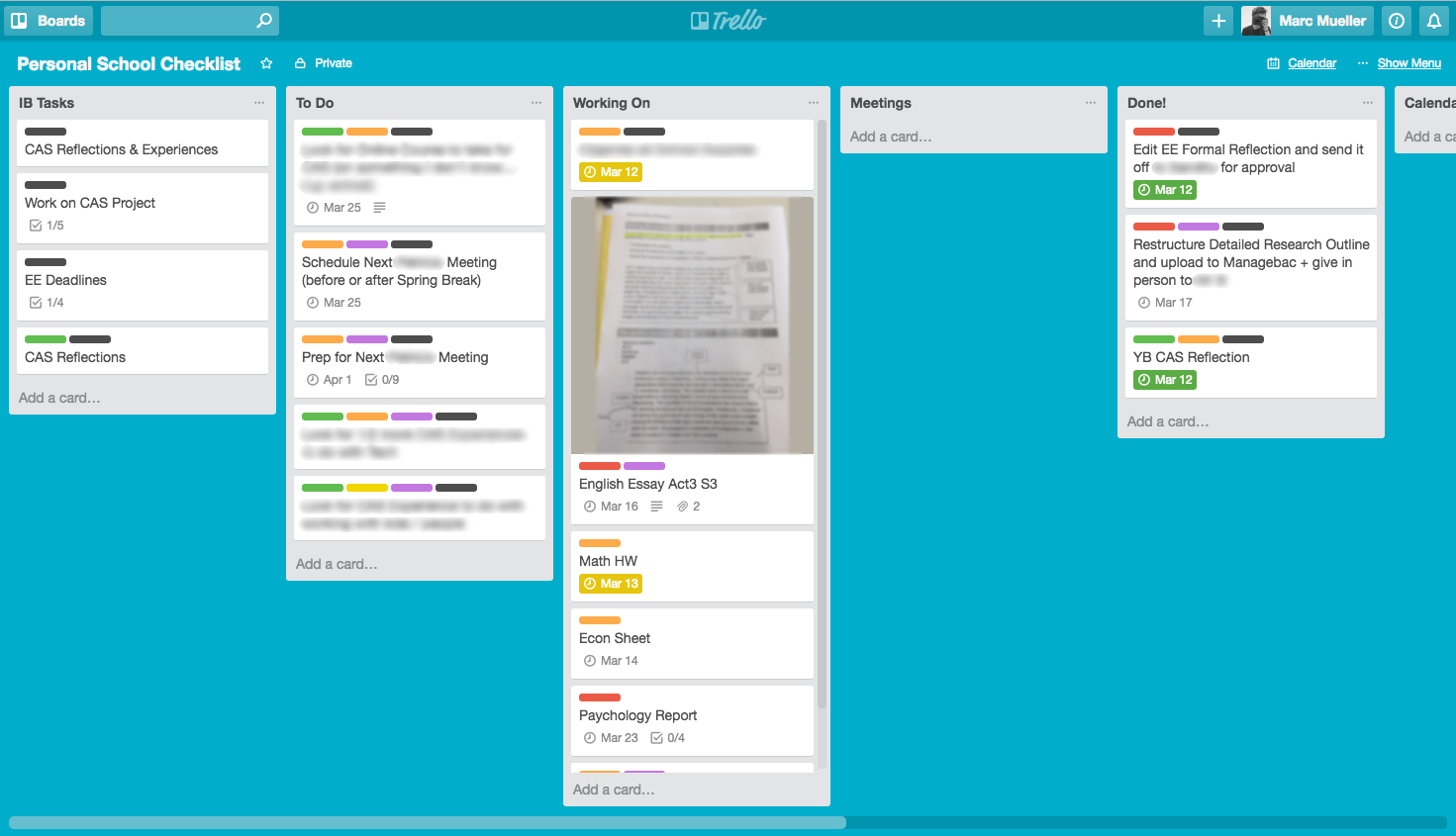
Such functionalities can be also achieved as a combination of multiple production tools, for instance I could set reminders and calendarize events using google calendar and encode the activities as tasks in productivity tool such as trello or similar

The following is a typical view of google calendar



while Trello could be used for tasks and status





Studybuddy will focus more on the ability to track the progress and make it really easy to stay on top of deadlines and upcoming assignments.

Tools like google calendar and Trello as much as other Microsoft productivity tools might still be of help for people so Study Buddy should provide account sync capabilities (not as part of the MVP being prototyped)

# Evaluation

Intro to be added here by Andreas

## Process

Andreas

## 

## Product

### End User Product Evaluation

In order to be easily accessible by the designated end users the application should be developed with a mobile first approach and using the material UI designs as much as possible.

Material UI has been selected as it has gained a lot of traction in the last few years and is being supported by multiple UI frameworks and this should avoid many usability issues as well as speed up the UI implementation.

Users might connect from several time zones, it is fundamental to receive notification in the relevant time zone for each specific user. The time zone should be set automatically according to the device being used to make the connection and/or geo-referencing the IP address from which the connection is being established.

Losing information could be really frustrating therefore the Study buddy application should allow the user to back up the information and store the backup in one of the user’s favorite cloud services or locally on the user’s machine.

As mentioned in the previous section, integration with 3rd parties application even though not being part of the MVP developments is a critical part of the application since many user expect to be able to export information and make them available to their favorite apps

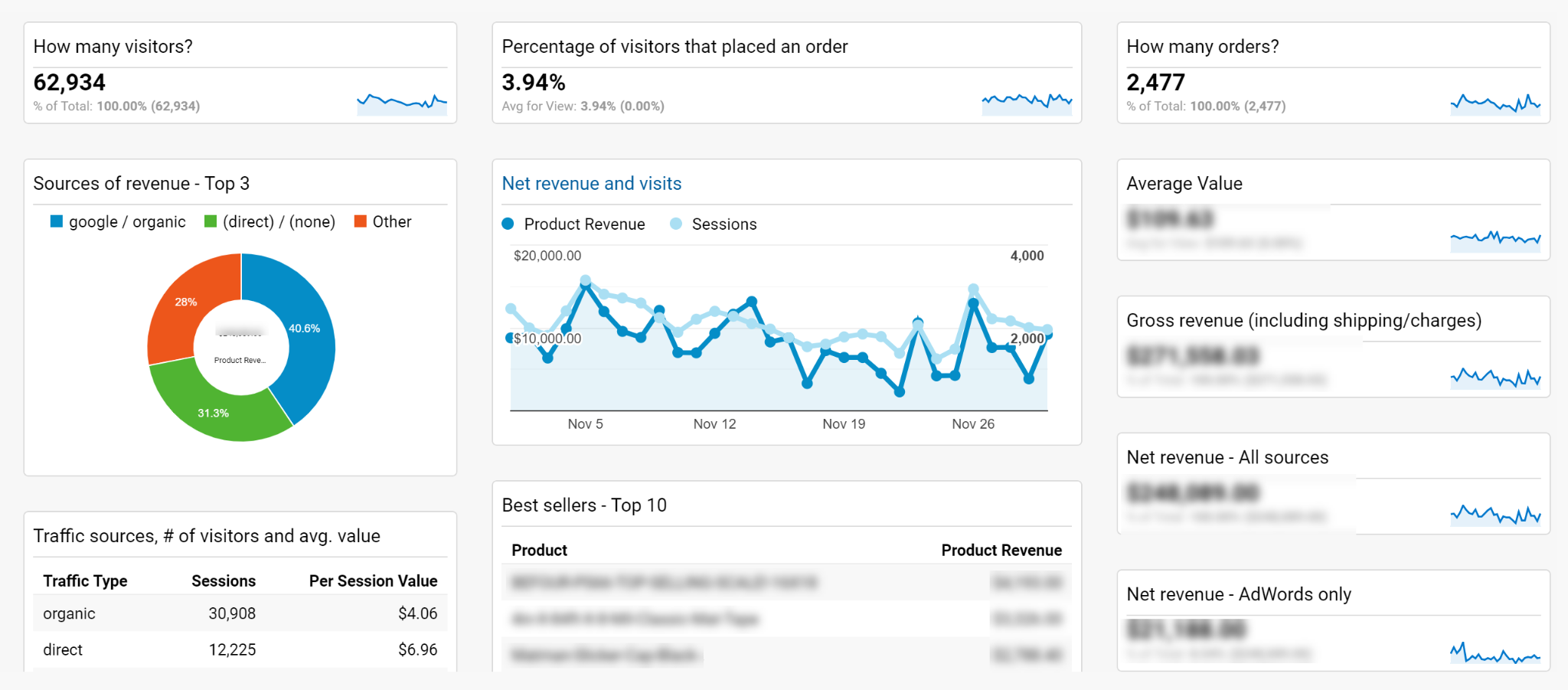
Implementing an open architecture and providing the ability to sync accounts with most used apps and networks should be part of the evolution plans.

### 

### Internal User Product evaluation

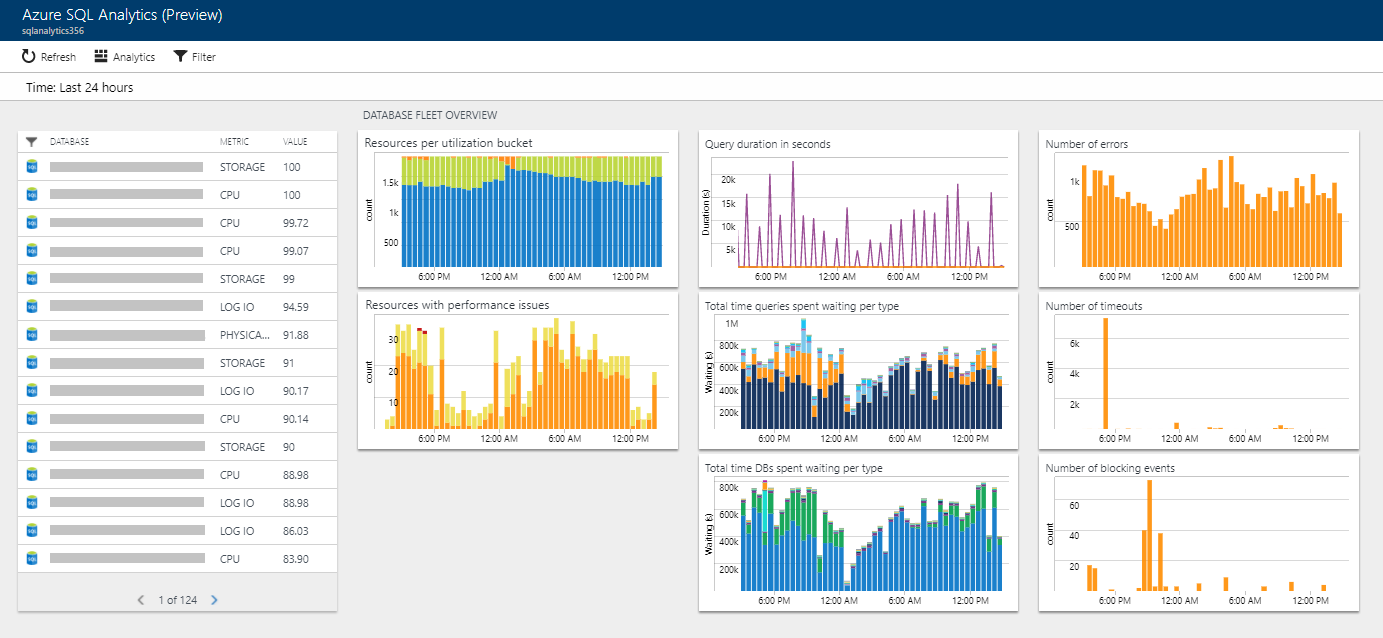
In order to understand how the application is being used overtime the application should be geared with tools for the analysis of traffic, being a web based application the best tool is probably google analytics

Below a screenshot



From a system perspective the application should be as available as possible to the end users even though it is in MVP therefore should be gered with a minimum set of monitoring services.

Azure, the cloud platform selected to build the prototype provides such services for the different tiers of the stack as shown below:



## Front-End

Swalin

## Back-End

Robert

## Testing

Jorge

// ONLY FOR INSPIRATION AND GUIDANCE. WILL BE DELETED.

You should evaluate your software development as both a set of processes (ie team working, methodologies employed) and the thing that you have built. Higher marks will be awarded where the evaluation is critical in nature, alluding to future work and highlighting areas for potential improvement.

---

8.103 Evaluating deliverables metrics

As a group, you should think about the following:

- what sort of metrics are you imposing to evaluate deliverables?

- are users tests sufficient or should you consider the wider context/implications?

- how close are you to meeting your aims specified in the proposal? Do your objectives match your current working process?

It is okay to reflect and change your working patterns based on feedback. What is important is that what you are measuring is both reliable and valid. These will be your markers for success and now might be a time in which you need to consider your direction.

---

8.202 Teamwork activity: designing and running tests

Task 1

In your team, design and run some tests on your software. You might want to use functional tests, user tests or a mix of both.

Task 2

Reflect on the feedback you have received from your tests. How can you conceptualise this under a broader set of constraints? Can you quantify or group the feedback in some meaningful way for analysis?

Consider aspects that work well and others that might need tweaking or changing.

Testing will make up a big portion of your report (and your marks) so test thoroughly! You might want to reflect back on Chapters 8 and 9 of the course textbook for this exercise.

---

9.104 Teamwork activity and deliverables: supporting people and keeping them safe

Task 1

In your team, think about your project:

- is there any potential for hurt or harm?

- what sort of demographic are you aiming for?

- are there any ‘at risk’ users?

- how might you design your software to support people and keep them safe?

- if the system were to cause harm to an individual, what do you think the personal and professional impact might be?

Try to conceptualise your ideas in some way. Perhaps you want to produce a table of these issues and then code them according to things like severity.

# Conclusion / Summary

Max

## Outcomes

The MVP can be used to test the initial idea and could be distributed among a community of students willing to test and provide feedback about the different functional and usability aspects.

## Development Practice

The Kanban was selected as the main instrument of collaboration and Trello as the technical tool to configure the activities and track the status of the activities as well as encode the details of such activities.

The team was able to perform two full scrum sprints and deliver a relevant number of functionalities including a number of change requests that have been added along the way while the application was being developed.

The team is composed of experienced individuals so the scope was tailored in order to be feasible within the available time frame and functionalities developed according to the initial plan.

We did not make the prescribed card estimation game but we preferred to leave the choice to each individual on which task to pick and we preferred to maximize the development capabilities in terms of development hours being executed by each member.

Since the resources were fixed in terms of team members and amount of hours to be made available by each we should have achieved the maximum amount of deliverables feasible with the given resources.

Many challenges were faced as a distributed development team, the first was the different time zones that made synchronous communication (such as MS Teams or Google Meet) available only for a short fraction of the day therefore we had to rely heavily on the capabilities provided by asynchronous communication tools such as chat (Slack), Kanban Task management (Trello), Code Versioning (Git) Document Management (Google Drive and Google Docs/Slides)

## Future Plans

The application has plenty of room for improvement in functional terms but it has been developed in a way that the initial set of features are solid enough to be built upon and not require a full rewrite in order to provide the desired evolutions.

The main area for evolution are related to integration with 3rd parties therefore making the app as part of a larger ecosystem

.

// ONLY FOR INSPIRATION AND GUIDANCE. WILL BE DELETED.

You should summarise the outcomes of your project and try to place it in a space. Here you might want to discuss where it fits in a dynamic workflow or the impact that it might have on a community of stakeholders or users.

---

10.103 Teamwork activity: reflecting on development practice

Task 1

In your team, reflect on your own development practice.

- How did your estimates match up to what really happened?

- What were the biggest challenges you faced as a group and how did you overcome them?

- Feedback to the group and talk about lessons you have learned from this exercise.

---

10.202 Future plans for your project

Of course, we all want to submit our work and get a good grade! But your project development does not have to end here. The work that you do throughout these modules should make up a portfolio that will help you to reflect on your development as a software engineer. Think about the following points:

- how might you utilise your software project in the future?

- do you plan on adding it to your portfolio?

- will you be continually refining your design in light of new feedback and knowledge generated?

Much like software, human beings develop over time. Our experiences, challenges and interaction modalities help us to learn, grow and develop!

# Appendix - Marking Rubric (To be deleted)

## Technical (30)

- Code solves challenges presented in the aims/objectives (4)

- Iterative design (4)

- Approach is fit for purpose without drastic oversights (4)

- Elegance/aesthetics/readability of code (2)

- Sensible approach to structure (e.g. index.html, style.css, functionality.js) (2)

- Evidence of collaboration and teamwork. Individual submissions should show engagement with a wide range of external resources and utilisation of tools such as peer review and discussion. (4)

- Evidence of milestones and reflective updates (e.g. updated Gantt charts, resource allocation.) (2)

- Difficulty of technical challenge (4)

- Novelty of technical challenge (4)

## Testing (10)

- All parts of the system working (2)

- UI evaluation (1)

- Well documented tests (1)

- Effective error handling (1)

- Systematic testing regime including design of appropriate test cases (1)

- Justification of testing methods (1)

- User evaluation involving representative stakeholders formative (2)

- User evaluation involving representative stakeholders summative (1)

## Report (30)

- Good introduction and fair discussion of literature (2)

- Clear statement of problem and effective problem analysis (2)

- Clear structure (1)

- Justification of design decisions (2)

- Good argumentation and justification of claims/problem analysis (2)

- Clear documentation and user guide (1)

- Sufficient and appropriate references, and good citing method (2)

- Good layout and formatting, especially of tables, figures, formulae and code example (2)

- Well written with systematic analysis/evaluation (4)

- Insightful discussion of results (4)

- Evaluation of own work in relation to original proposal and plan (4)

- Conclusion and discussion of future work (4)

# Appendix II - Brainstorming & Notes (To be deleted)

To-do List

* Discuss team principles (Agile-like) (?)
* Decide Kanban/Scrum or other and discuss why (cite literature)
* Assign Roles (PO, UX/FrontEnd, Tech/BackEnd, PM, Testing)
* Choose Tools
  + Website Hosting - Azure
  + Database Hosting - Azure
  + Team Communication - Slack i/o MS Teams
  + Version Control - GitHub
* Choose Technologies
  + Database = MySQL or MS SQL (Robert to write report on what and why)
  + Back-End = REST API .NET (Robert to write report on why)
  + Front-End = HTML / Bootstrap CSS / JS / jQuery / JSON (Swalin/Andreas to write report on what and why)
* Create Backlog (Not Started, In Progress, Testing, Done) and include Acceptance Criteria (Max/Andreas) (WIP)
* Design Sprints (Max/Andreas) (WIP)
* Risk Assessment (note to self: Robert-Specific-Technologies) (Andreas to write report on Risk Assessment)
* Man Pages (Tech for running the MVP and User Manual) (Jorge to write man pages)

Kanban - Not started -> In Progress -> Testing -> Done

Backlog creation

Roles Assignment (Developers, Scrum Masters, Product Owner)

Sprint

Architecture (Database, Backend, Frontend)

Choice of tools (Documentation comparison)

Risk Mitigation on choice of tools (one knows MS stuff)

PO: Max

Scrum Master: Jorge

Developers: Robert/Andreas/Swalin

# Appendix III - Backlog & Sprint Planning (WIP)

1. As a student/user I want to be able to register and create login credentials, **works when** I am able to provide an email and I am then guided through the selection of the password by providing a minimum of 8 characters, containing one capital, one number and a special character. Last step, I want to receive a confirmation email that allows me to activate the account.
2. As a user I want to be able to log into my created account, which is linked to the email address I provided which **works when** I use my registered email address and selected password and I can see my personalised dashboard.
3. After the first log in the user should be prompted to start adding modules, **works when** after the main page is reached for the first time the user is shown a message inviting to set up his/her first module by providing the minimum set of information i.e. Title and Short Description.
4. For each module added the user should be able to set events such as deadlines to be notified, **works when** for each module created the user has access to an editing functionality that allows to add event based information along with details about notification rules.
5. As a user I want to be able to easily change the progress of my modules, **works when** either form the main page or from the module details page I can change the progress through a fader
6. As a student i want to be able to search easily through search engine and display list with all Modules, **works when** user enter the search term and the user can further filter the list based on various parameters as this benefits the user in reducing searching time
7. As a student, each request I submit should be processed within 10 seconds on the application.
8. As a student, I want to receive Emails with a latency of no greater than 2 hours.
9. As an Impatient student, I want to load the website within 3 seconds when the number of simultaneous users is greater than 1,000.
10. As a student, I want to be able to run your study tracking application on the latest version of Windows.
11. As a student, I want the site to be available 99.999 percent of the time when I try to access it, so that I don't get frustrated and find another site to use.