

COMP[39]900 Term 2 2024 Project Proposal Assessment

- ◆ **Project Proposal** (due **Week 3 Sunday (16 June) @ 9:00pm**) (worth 10%)
- ◆ Make sure you also look at the Moodle Proposal assessment submission link for submission instructions. Follow those submission instructions.
- ◆ The project proposal should be self-contained. i.e. **no content should be outside of the report and simply linked to.**
- ◆ The project proposal must follow the following **formatting requirements**:
 - Include a title page containing course code, course title, project number and title, team name, each member's name, email, student ID, role, and project proposal submission date.
 - Be at least 10 pages long using at most 12pt font with reasonable margins and spacing, not including the title page, the table of contents, and references page, and be in PDF format.
 - Include a table of contents and page numbers.
 - Include full references and in-text citations. Use either APA referencing style (<https://student.unsw.edu.au/apa>) or Harvard referencing style (<https://student.unsw.edu.au/harvard-referencing>).

The main purpose of your proposal is to

1. Identify and research the problem you are solving.
2. Propose a solution based on the research you perform and justify how your solution is unique and overcomes problems you identify in existing systems.
3. Set out the project management that you will use for the duration of the project (user stories, sprint milestones, project objectives).

Marking for this proposal is based on how well each of the above aspects is covered. More specifically, your proposal must include:

a) Background (15%)

- a. Clearly identify the problem domain and problem(s) being solved. Add a brief summary of your solution.
- b. Discuss **at least two existing related works or systems** in your identified problem domain. For each related work, make sure you:
 - i. Describe the problem their system solves.
 - ii. Identify and discuss any strengths of their work that you will take into consideration in your design.
 - iii. Identify and discuss any weaknesses or limitations of their work that you will avoid or overcome with your design.
 - iv. Make sure to reference all works you discuss following the recommended referencing guidelines.

b) User stories and sprints (25%)

- a. Include a product backlog of correctly structured user stories, describing the functionality to be delivered, with **screenshots** showing all these user stories defined in **Jira**. The entire text of each user story should be readable inside the report.

- b. Identify the **user stories** in scope for the **first sprint** with **screenshots** showing all user stories allocated to the first sprint in **Jira**. The entire text of each user story should be readable inside the report.

To satisfy this criterion, you must **allocate a realistic amount of work** to your first sprint. If you have too few user stories allowed to the first sprint, you risk delaying the completion of essential features to stakeholders.

What is a realistic amount of work?

This is heavily dependent on your group and the chosen project.

Although this is not heavily weighted in this assessment, it will help you greatly in a few weeks if you plan out work estimates well now.

There are many ways you can do this:

- i. Assign story points to each user story on Jira and then allocate roughly a third of the story points to the first sprint.
- ii. Plan your sprint milestones and align the selected user stories selected with each milestone. If you chose to do this, make sure that each milestone represents roughly the same amount of work.
- iii. Create smaller deadlines for a set of features and then align user stories with these deadlines such that you maintain a consistent amount of work during the term.

Estimating the time required to complete user stories is an important team skill and it is important to commit to a demanding but manageable workload. Committing to too little work results in your final product will not be as developed as it could be. Overcommitting means you won't finish all your work by its deadline. Both of these scenarios will be taken into account in all assessment tasks within this course.

- c. Defines the **start and end dates** for all **sprints** envisaged during the term. Briefly describe your milestones for each sprint.
 - i. Note: ensure that the sprints you define allow you to undertake a **progressive demo** in each of **Weeks 5 and 8**; as well as a **retrospective** before resp. the end of **Weeks 5 and 8**. Hence, **three sprints** will be ideal.

c) Technical Design (55%)

- a. Include a **system architecture diagram** that shows the interactions between the core systems of your application. This can be in the form of a **UML diagram**, but a **simpler diagram** containing boxes with object names and **labelled** arrows between them can be just as effective.

For web-development and other development focused projects, make sure your diagram clearly demonstrates:

- i. how different components of the backend interact with each other.
- ii. how the backend communicates with the frontend.

- iii. how your software communicates with external databases, APIs, and other services.

For machine-learning and research projects, make sure your **diagram** clearly demonstrates the **interactions between your models**, the transfer of data to and from where it is stored, how external components such as a user interface interact with a backend that processes requests, and how the backend manages different models.

Note that an effective system architecture diagram **should not be overly complex** and each component within the diagram should only be responsible for a small set of uses. If you find that one component of the backend is handling multiple tasks (calls to the database, parsing information from the frontend etc), **consider breaking it down into multiple smaller components**.

Some recommended websites to develop these are **Draw.io** (<https://draw.io/>) and **LucidCharts** (<https://lucid.app/>).

b. **OPTION 1 (for development projects):**

Develop **storyboards** to illustrate all major system functionality. Mention which **user stories** are covered by each **storyboard** (one storyboard can encompass multiple user stories). Not all user stories need to be covered but **all major functionalities must have storyboards**. The most **effective storyboards** will have the following:

- i. Showcase what the frontend interface will look like, including colour schemes and layout.
- ii. Visually demonstrate how to navigate to different parts of the application.
- iii. Include boards showing how each major feature works and is intended to be used by users.

The requirements for your project will constantly evolve throughout the term but with well-researched storyboards, it is highly likely that the majority of features in your project will **closely resemble** your storyboards.

Some recommended websites to develop these are **Figma** (<https://www.figma.com/>)

OPTION 2 (for machine-learning and research projects)

Conduct a **literature review** to determine the current research methods relevant to your project. This is **completely separate** to the background section of your proposal and should be much more **focused on**:

- i. what research exists related to your chosen project.
- ii. the novel methodology used in this research and how you build on top of that with your own planned project.
- iii. the evaluation metrics used to evaluate their models.
- iv. the limitations of prior models and how you plan to overcome these.

There might not be any research in your specific problem domain. If that is the case, you should explain why. Is it a unique problem that has only arisen recently? Has there been a new technology developed recently that has enabled new development? Note that even if this is the case, you must still conduct a thorough literature review on adjacent research areas that will help you develop your own work. It is also highly recommended that you add a **proof of concept** or some **storyboarding** that showcases how you envision your research being used and what the outcome of your project will ideally look like.

One good starting point for your literature review is to search for your topic on Google Scholar (<https://scholar.google.com/>) to find well-known papers online in your problem domain. Look at the literature review section of these papers. It is also very useful to read survey papers that establish and explain the state of the art within a specific problem domain. To achieve a high mark for this, you should thoroughly cover **3-5 research papers**.

You can make a judgement on which option is most appropriate for your project.

- c. Include a list of **design justifications** for your planned solution. This should include:
 - i. Breaking down the problem you have chosen into more **manageable subproblems**.
 - ii. A **summary** of how you plan to solve each subproblem.
 - iii. For the more complex subproblems, **brainstorm alternative solutions** and **justify** why you chose the solution you did rather than the alternatives.

How you format these is up to you, but effective design justifications tend to incorporate research from the background section, identify how decisions will affect users and identify possible limitations in your design and how you will mitigate these. **Also discuss how your solution provides novel functionality beyond existing systems wherever relevant in your design justifications.**

- d. Clearly communicate how all project **objectives/functionalities** are satisfied by your **user stories**.

Note that to achieve the maximum mark for each of these criteria in this section, you must not only include what is listed but also **demonstrate research, planning and effectively structure your proposal.**

d) Report Quality

- a. The report should conform to the formatting requirements in this specification.
- b. The report should be easy to read and concise.

A detailed marking rubric can be found on the next page.

Category	Max Mark
Background (15%)	1.5
Clearly identifies the problem domain and problem(s) being solved	0.5
Clear evidence of research into existing systems in the same problem domain and understanding of the benefits and limitations of current software solutions	1
User Stories and sprints (25%)	2.5
Product backlog of correctly structured user stories, describing the functionality to be delivered, with screenshots showing all these user stories defined in Jira	1
Identifies user stories in scope for the first sprint with screenshots showing all user stories allocated to the first sprint in Jira	1
Defines the start and end dates for all sprints envisaged during the length of term. Identifies sprint milestones for each sprint	0.5
Technical Design (55%)	5.5
Effective system architecture diagram	1
Effective use of storyboarding to illustrate the layout of the application and all major functionalities OR Effective literature review covering 3-5 research papers establishing current work in the problem domain, evaluation metrics and limitations	2
Detailed outline of subproblems, list of design justifications, novelty of the solution and the mitigation of limitations	2
Clearly communicate how all project objectives are satisfied by the user stories that are defined	0.5
Report Quality (5%)	0.5
Report conforms to the specified formatting requirements and is easy to read	0.5
Total Mark	10