3. Project methodology and outcomes

3.1. Initial project plan

3.1.1. Tasks and milestones

Category	ID	Task/Milestone Details	Priority	Est. Duration
Planning	1.1	Define a set of aims and objectives to inform the project direction.	High	3 days
Planning	1.2	Consider any causes for ethical concern. Document potential risks and appropriate mitigations.	Medium	1 week
Planning	1.3	Break down and sequence project tasks to produce a Gantt chart of the project schedule.	Medium	1 week
PDD	2.1	Write a section covering the project aims, objectives and scope. Define a set of deliverables.	High	3 days
PDD	2.2	Write a section conveying the benefits of the project and how it will be conducted.	Medium	3 days
PDD	2.3	Ask for feedback on current PDD draft, then make revisions.	Low	1 day
PDD	2.4	Detail an account of how the project will be undertaken in terms of tasks and scheduling.	Medium	3 days
PDD	2.5	Ensure that the document is well-presented and coherent.	Low	3 days
PDD	M-A	**14th Oct, '24** - The Project Definition Document (PDD) has been completed.	Milestone	3 weeks
Report	3.1	Write a section introducing the project including its aims, objectives and context.	Low	1 week
Report	3.2	Conduct a literature search, reading papers and building an overall narrative for the review.	High	2 weeks
Report	3.3	Read relevant articles and write about metaheuristics and their applications.	Medium	3 weeks
Report	3.4	Read relevant articles and write about the history of MSA software and approaches.	Medium	3 weeks
Report	M-B	**25th Oct, '24** - The literature review has been completed.	Milestone	5 weeks
Report	4.1	Capture a discrete set of requirements to direct the design and development of the tool.	High	3 days
Report	4.2	Produce UML diagrams as high-level software design incl. Use Case, Package, Class diagrams.	Medium	3 days

Figure 3.1.1a *– A table of tasks & milestones covering project planning, report writing and software design. (Part 1 of 2)*

Category	ID	Task/Milestone Details	Priority	Est. Duration
Report	4.3	Ask for feedback on literature review and design sections, then make revisions.	Low	2 weeks
Report	4.4	Write a technical background section on the MSA problem to provide context for readers.	Low	2 weeks
Report	4.5	Evaluate the project as a whole, including comments on development and software testing.	High	2 weeks
Report	4.6	Ask for feedback on technical background and evaluation sections, then make revisions.	Low	2 weeks
Report	M-J	**1st Apr, '25** - The final report has been completed, giving a detailed account of the project.	Milestone	6 months
Soft. Dev.	M-C	**12th Nov, '24** - MAli v0.1 is released - a basic tool that produces low quality, but valid solutions for MSA.	Milestone	2 weeks
Soft. Dev.	M-D	**26th Nov, '24** - MAli v0.2 is released - introducing a metaheuristic algorithm to guide the alignment.	Milestone	2 weeks
Soft. Dev.	M-E	**10th Dec, '24** - MAli v1.0 is released - an improvement on v0.2 and indicative of a full implementation.	Milestone	2 weeks
Soft. Dev.	M-F	**14th Jan, '25** - MAli v1.1 is released - resulting from experimentation on the v1.0 design.	Milestone	2 weeks
Soft. Dev.	M-G	**28th Jan, '25** - MAli v1.2 is released - producing a selection of high-quality solutions as output.	Milestone	2 weeks
Soft. Dev.	M-H	**11th Feb, '25** - MAli v1.3 is released – fulfilling the defined requirements as a capstone of development.	Milestone	2 weeks
Evaluation	5.1	Test the defined functional & non-functional requirements using a university desktop computer.	High	1 week
Evaluation	5.2	Test the tool's performance relative to available alternatives using structural benchmarking.	Medium	1 week
Evaluation	5.3	Analyse the performance and quality of the tools tested in 5.2. Discuss the results.	Medium	1 week
Evaluation	M-I	**4th Mar, '25** - Software requirements have been tested and a case study has been completed.	Milestone	3 weeks

Figure 3.1.1b *– A table of tasks & milestones covering report completion, software releases, and evaluation of the project. (Part 2 of 2)*

A Note on Development Work Items

The development of the software will be undertaken with the agile principles (Beck et al., 2001) in mind. In order to be open to change, a prioritized backlog of work items has been provided in place of definitive tasks (see **Appendix b**). This backlog will likely be revised and expanded over the course of development, to accommodate new priorities identified from research and ultimately produce a more valuable product.