Development Plan
ProgName

Team #, Team Name
Student 1 name
Student 2 name
Student 3 name
Student 4 name

Table 1: Revision History

Date	**Developer(s)**	**Change**
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

[Put your introductory blurb here. Often the blurb is a brief roadmap of what is contained in the report. —SS]

[Additional information on the development plan can be found in the lecture slides. —SS]

1 Confidential Information?

[State whether your project has confidential information from industry, or not. If there is confidential information, point to the agreement you have in place. —SS]

[For most teams this section will just state that there is no confidential information to protect. —SS]

2 IP to Protect

[State whether there is IP to protect. If there is, point to the agreement. All students who are working on a project that requires an IP agreement are also required to sign the "Intellectual Property Guide Acknowledgement." —SS]

3 Copyright License

[What copyright license is your team adopting. Point to the license in your repo. —SS]

4 Team Meeting Plan

[How often will you meet? where? —SS]

[If the meeting is a physical location (not virtual), out of an abundance of caution for safety reasons you shouldn't put the location online —SS]

[How often will you meet with your industry advisor? when? where? —SS]

[Will meetings be virtual? At least some meetings should likely be in-person. —SS]

[How will the meetings be structured? There should be a chair for all meetings. There should be an agenda for all meetings. —SS]

5 Team Communication Plan

[Issues on GitHub should be part of your communication plan. —SS]

6 Team Member Roles

[You should identify the types of roles you anticipate, like notetaker, leader, meeting chair, reviewer. Assigning specific people to those roles is not necessary at this stage. In a student team the role of the individuals will likely change throughout the year. —SS]

7 Workflow Plan

- How will you be using git, including branches, pull request, etc.?
- How will you be managing issues, including template issues, issue classification, etc.?
- Use of CI/CD

8 Project Decomposition and Scheduling

- How will you be using GitHub projects?
- Include a link to your GitHub project

[How will the project be scheduled? This is the big picture schedule, not details. You will need to reproduce information that is in the course outline for deadlines. —SS]

9 Proof of Concept Demonstration Plan

9.1 Main Risks for Project Success

The most significant risks for our Buddhist manuscript fragment reconstruction platform are:

- **Inconsistent OCR Performance**: OCR accuracy varies dramatically across different manuscript conditions, script styles, and image quality. Ancient Buddhist texts often contain deteriorated characters, non-standard orthography, and multiple script variations that challenge standard OCR engines.
- **Lack of Positive Training Examples**: The absence of confirmed fragment matches creates a training data challenge. Without verified examples of fragments that belong together, we will need to explore unsupervised learning approaches and develop alternative validation strategies for our matching algorithms.

These are risks because they directly impact the core functionality of our system - the ability to accurately identify which fragments belong together.

9.2 Implementation Challenges

The most challenging aspects of implementation will be:

- **OCR Integration and Consistency**: Implementing OCR engines that can handle Sanskrit manuscript variations while maintaining consistent output across different image conditions and script styles.
- **AI Model Training Without Ground Truth**: Creating and training machine learning models for fragment similarity matching without access to verified positive examples of matched fragments. The final model will need to combine multiple features (OCR text, edge patterns, damage signatures) into confidence scores, which presents challenges for training validation. We plan to explore unsupervised learning approaches to address this challenge.
- **Interactive Canvas**: Building a responsive, intuitive drag-and-drop interface that can handle potentially hundreds of fragment images while maintaining performance.
- **Multi-scale Matching**: Implementing algorithms that can suggest matches at different confidence levels.

9.3 Testing Difficulties

Testing will present unique challenges because:

- **Ground Truth**: We need access to known fragment matches from Buddhist Studies scholars to validate our algorithms, which may be limited initially.
- **Subjective Validation**: Fragment matching often involves scholarly interpretation, making automated testing metrics challenging to define.
- **Performance Testing**: Testing the system with large batches of high-resolution fragment images will require substantial computational resources.
- **User Experience Testing**: The interface must be intuitive for scholars with varying technical backgrounds, requiring extensive usability testing.

9.4 Library and Technology Risks

9.5 Hardware and Infrastructure Concerns

- **Processing Power**: Computer vision and AI algorithms may require substantial CPU/GPU resources for model training.

9.6 Demonstration Plan

To address these risks and demonstrate feasibility, our POC will focus on proving that the core technical challenges can be overcome:

- **OCR Functionality Demonstration**: Show OCR engines successfully extracting text from sample manuscript fragments with varying quality levels. We will demonstrate text extraction from both clear and degraded fragments to validate OCR consistency approaches.
- **Image Segmentation and Normalization**: Demonstrate fragment segmentation algorithms that can isolate individual fragments from composite images and normalize them for consistent analysis (rotation correction, standardization).
- **Basic Feature Extraction**: Show extraction of key features from fragments including edges, text content, and damage patterns that could be used for matching.
- **Prototype Similarity Metrics**: Implement basic similarity algorithms that compare normalized fragments and provide confidence scores, acknowledging the limitation that it will be difficult to validate accuracy without ground truth data.
- **Interactive Workspace**: Create a basic interface where users can upload fragments, view OCR results, and see preliminary similarity suggestions.

This demonstration will prove the technical feasibility of the core components. The POC focuses on demonstrating that our technical approach can extract and compare the right types of features for fragment matching. While the absence of ground truth data presents challenges, we will explore unsupervised learning techniques and develop validation approaches that can work with the available data.

10 Expected Technology

[What programming language or languages do you expect to use? What external libraries? What frameworks? What technologies. Are there major components of the implementation that you expect you will implement, despite the existence of libraries that provide the required functionality. For projects with machine learning, will you use pre-trained models, or be training your own model? —SS]

[The implementation decisions can, and likely will, change over the course of the project. The initial documentation should be written in an abstract way; it should be agnostic of the implementation choices, unless the implementation choices are project constraints. However, recording our initial thoughts on implementation helps understand the challenge level and feasibility of a project. It may also help with early identification of areas where project members will need to augment their training. —SS]

Topics to discuss include the following:

- Specific programming language
- Specific libraries
- Pre-trained models
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Tools you will likely be using?

[git, GitHub and GitHub projects should be part of your technology. —SS]

11 Coding Standard

[What coding standard will you adopt? —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. Why is it important to create a development plan prior to starting the project?

2. In your opinion, what are the advantages and disadvantages of using CI/CD?

3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

Appendix — Team Charter

External Goals

Our project will contribute to the field of religious studies by providing valuable manuscripts that have been lost due to the fragmentation of the texts. We will also develop our knowledge of machine learning and computer vision through this project which we will helpful for our future professional development. Our tool will be able to be used by anyone studying religious studies so that they can create their own texts, which will lower the barrier required to provide valuable information to the field.

Attendance

Expectations

Our team expects all team members to arrive on time for team meetings unless they are late due to an acceptable excuse. All team members must be present for the entire meeting unless they have an reason for leaving early. Team members are expected to be prepared for the meeting and participate in the discussions to provide their viewpoint on the topics. Team members are expected to attend every meeting unless an acceptable excuse is given.

Acceptable Excuse

An acceptable excuse for missing meetings or deadlines includes family emergencies such as a death in the family or serious illness. Medical emergencies such as illness or doctor appointments will also be acceptable. If there is a conflicting class or midterm/exam taking place during the meeting, the team member will be allowed to attend the event, but we will try to avoid planning meetings during these events.

Poor time management or missing meetings due to other commitments will not be tolerated, and team members are expected to notify the entire team about their reason for missing the meeting as soon as possible so that the team can plan in advance.

In Case of Emergency

In an emergency situation, team members must notify the team and provide the emergency details. They must coordinate with the team to fill the gap in coverage so that the work will be able to finished. They must let the team know the approximate length of time required for the emergency, or keep the team updated during the emergency so that they can plan ahead if necessary. If possible, provide any work that is in progress to the team so that they can continue working on it. If this emergency happens over a long period of time, we will notify the instructor.

Accountability and Teamwork

Quality

All team members must come to meetings with meeting topics and questions prepared, and all necessary tasks must be completed prior to the meeting. All team members must also read the agenda and meeting notes. Our documentation must satisfy the course requirements and be easy to understand, so that we can easily reference it. We aim for our code to have high readability and maintainability. Furthermore, our code will be tested rigourously through unit testing for basic coverage and regression testing to ensure that the new code does not cause any defects, and all code will be code reviewed to reduce technical debt. We will also ensure that our deliverables are cited properly and peer reviewed.

Attitude

Our team must communicate in a respect manner with each other, and include everyone in group discussions. We will work collaborately by providing feedback to improve the quality of the work and through knowledge sharing. We will also focus on having a positive attitude when dealing with problems and we will be receptive to new ideas to maximize our creative potential.

Code of Conduct:

- Effectively communicate and collaborate with each other
- Be respectful to all team members
- Avoid using personal attacks or harsh language
- Violating the code will result in further consequences

Stay on Track

We will incentivize good performance by keeping track of metrics such as the number of pull requests, so team members feel more motivated to increase their number of pull requests. Furthermore, we will keep track of how team members are performing outside of the milestones which includes team member roles such as creating the agenda and meeting minutes. Some performance targets includes participation in all meetings, completion of tasks before the deadline and participation in code reviewing and giving feedback. If any team members are underperforming, we will first discuss this in a meeting to ensure that they are aware that they are not performing well. If their performance does not improve, we will bring this issue up in a TA meeting so that the TA can give feedback to the team member. If the team member is still not performing well, we will get the instructor involved in the discussion. Finishing your tasks early will give team members the opportunity to provide more feedback to others, and they will be able to get started on the next milestone earlier.

Team Building

We will build team cohesion from establishing a friendly team environment by discussing topics unrelated to the project occasionally and by getting to know each other. Furthermore, we will acknowledge each other accomplishments and promote working together on problems to build team chemistry. Team members will be encouraged to spend time together outside of time spent working on the project. Also, since we are enrolled in the same courses, team members will be encouraged to collaborate on team projects or labs in other courses.

Decision Making

Since our group has 5 team members, we are able to make decisions based on majority vote. The team leader will be responsible for mediating during disagreements, and we will keep voting anonymous to avoid bias affecting the vote. Before the vote, we will have a discussion involving all group members, and we will not be able to vote if all team members are not present, to prevent team members from being left out. All team members will be able to bring up topics for decision making, but the team leader will be responsible for when the vote will be held to avoid spending too much or too little time making a decision.