The Team

Release Plan

E-Tendance - Facial-Recognition Based Attendance Taking System

[Release Plan]
Version 1.1 Approved

The Team

Li Shanlan
Zeng Jinpo
Akshaya Muthu
Simon El Nahas Christensen
MN Shaanmugam
Cao Ngoc Thai

Version History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Simon	11/08/2019	Harry	11/11/2019	Initial Release Plan draft
1.1	Harry	11/11/2019	Simon	11/13/2019	Final Edits

1. INTRODUCTION

The release plan is a detailed description of the functionality E-Tendance of the team application, and the future release date versions. The purpose of this release plan is to help the group identify the goals of the project and track the advancements and improvements closely. By following the plan, the criteria for each release version can be identified based on the importance or urgency of the task and the implementation of the CI/CD (continuous integration/continuous development) practice.

2. REFERENCED DOCUMENTS

Version #	Retrieval date	Document
1.0	8. Nov 2019	System Requirement Specifications (SRS)
1.1	8. Nov 2019	Project Plan
1.1	8. Nov 2019	Risk Management Plan

3. OVERVIEW

E-Tendance is built based on the need for an automated attendance taking system in schools and universities. As traditional approach is manual intensive and error prone, a more automating, facial recognition supported approach is required.

At the early state of the development process, UCD with use case diagrams are written. In the next step, SRS shows all functional and nonfunctional requirements needed for the system. Risk management and project plan then written to guide the development process. Test Plan then written as a guideline for quality assurance process.

The final system architecture has evolved from the initial design as below. Most significant changes are utilising in-house facial recognition service, implement reverse proxy for scalability.

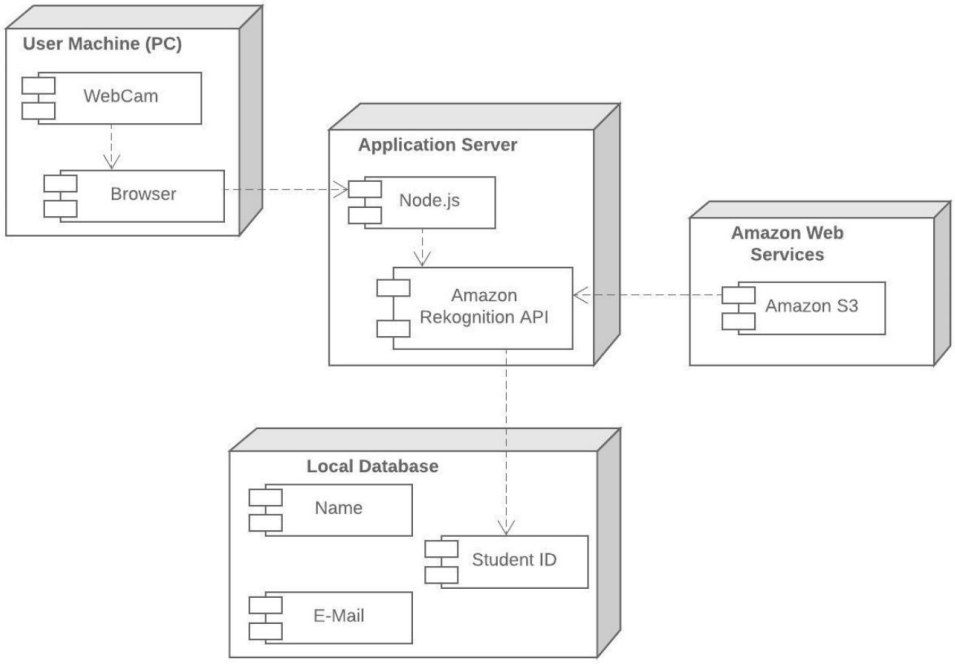

Figure: Initial System Architecture

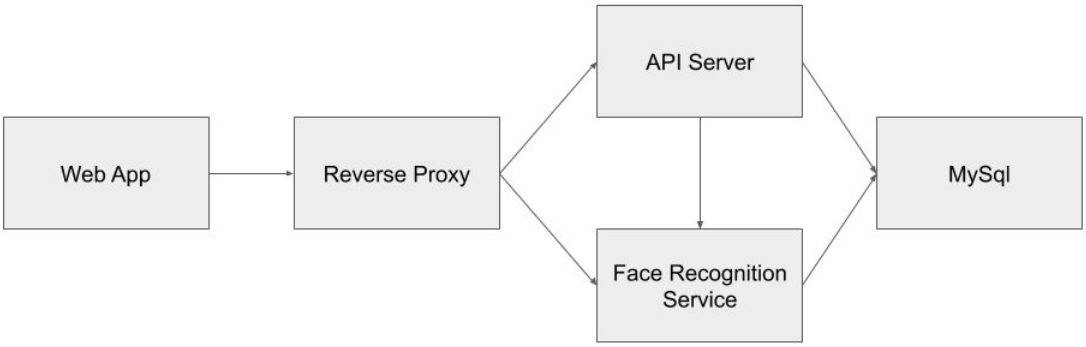

Figure: Final System Architecture

4. ASSUMPTIONS,CONSTRAINTS,RISKS

4.1. Assumptions

4.1.1. Development

The availability of development computer resources will be available throughout the project

4.1.2. Staff allocation

The allocated staff of 6 people will stay on the project from hiring and shoot off until the end of the project and will deliver the contracted working hours.

4.1.3. Resource allocation

the allocated resources in terms of salary for staff members will be sufficient for the work required throughout the project period.

4.1.4. Environment

The environment for which the project is developed will remain constant throughout the project period

4.1.5. API

Any APIs that might be used throughout the project will not change their interface and availability.

4.2. Constraints

4.2.1. Work Force

The work force and the capable work load will only total to a finite amount that will also have to be spent on other crucial aspects of the project. e.g. Implementation, documentation, testing etc. Therefore it is important to allocate the appropriate amount of this work force to the release cycle. If at some point during the project the staff working on this estimates that there will be a need for more work force then they are responsible for communicating that to the team leader, which will then consider to reallocate some of the work force.

4.2.2. Resources

The monetary resources to develop the project is finite and so any activities that can cause excessive use of the monetary resources should be identified. These could include use of expensive software or hardware. For this reason throughout the project staff should seek to use free alternatives and optimise the resource usage. If at some point during the project the staff working on this estimates that there will be a need for more money having already tried alternatives that didn't show to be sufficient. Then they are responsible for communicating that to the team leader, which will then consider to reallocate money from other places in the project.

4.2.3. Deadline

The project is bound by the strict deadline of the end of the project. This will is not to be changed and therefore it is necessary that each staff member will report and the project leader identify slippage. Once identified, the project leader will seek to decrease the scope of the task and report such decisions to the customer.

4.3. Risk List

Risk	Mitigation strategy
Important staff are unavailable	Each component is assigned at least 2 members, so that there will always be one to cover for the other, when the other one is unavailable
Cost Overrun	Cost control during the implementation, and remove unnecessary expenses during the implementation. Also revising the project schedule and look into available crash weeks.
System not meeting the requirements as discussed	Weekly meeting to clarify the requirements, and gather user feedback
During the project implementation phase, unidentified requirements emerged.	Ensure that consumers and developers are present together at each point of the specification review. Ensure that agile implementation is pursued so that, when specifications are applied to the process during each sprint session, the development is not hampered.

Computer or hardware glitches or faults that may cause scheduling delay.	Hardware / software error testing must be reported and maintained during each step of the lifecycle of the project.

5. RELEASE APPROACH

5.1. Rationale

As detailed in the project plan the SDLC (Software Development Life Cycle) is based on developing iteratively and delivering a final product. Following lehman's law on continuing change: “the software must be continually adapted or it becomes progressively less satisfactory”[1]. To accommodate for the continual adaptation the following release strategy is devised for the life cycle of the software product after it has been released and is in production.

This entails both correcting bugs and continuing adaptation for the change in environment that inevitably happen as time passes.

5.2. Release Strategy

The release of E-Tendance has been split up in parts small enough for meaningful for sprints. The sprints has then been structured in such a way that the functionalities that were most crucial was tackled first. iteratively the sprints would then be tracked to combat any issues early on.

The most expensive development phase will be the for the first few iterations. The core functionalities of the software application. All Core functionality will be delivered in this Release 1.0.0.

The emailing of students will be delivered later on in separate minor release v1.1.0

Depending on the need for urgent fixes 1.0.1 the release plan will be agile in the sense that it will accomodate the need for each of such releases. The release plan specifies here that the fix releases 1.0.1-3 will be delivered every week for the following three weeks after the release 1.0.0. To get the fixes out to the customers as soon as possible.

[1] Lehman, M. M. (1980). "On Understanding Laws, Evolution, and Conservation in the Large-Program Life Cycle". Journal of Systems and Software. 1: 213–221

Three weeks after release 1.0.0 the fix releases will be released if needed on a monthly basis.

New major or minor releases will only be released when the need is encountered. And a new project will be devised for this process in agreement with developer and customer.

5.2.1. Release Content

Release v1.0.0	
7.1 Facial Recognition The system must automatically and accurately detect and recognise students' appearance in the class based on the students' uploaded photos.	Functioning as specified in SRS
7.2 Attendance-Taking The system must update the database to mark a student as present, late or absent for a class. A real-time attendance status is displayed on the screen for the current class.	Functioning as specified in SRS
7.3 Unrecognized Faces In the case of unrecognised faces, the system must prompt students to input either retake attendance, or take additional photos of themselves and update the database.	Functioning as specified in SRS
Release v1.0.1-3	fix release every week for the following three weeks and after that if needed on a monthly basis.
Release v1.1.0	

7.4 E-Mail Students The system will automatically disseminate a batch of emails informing absent students. The email contains the details of the class and the actions that students need to take, such as submitting an MC.	Functioning as specified in SRS
Release v1.1.1-3	fix release every week for the following three weeks and after that if needed on a monthly basis.
Release v1.2.0	new functionality or maintenance due to environment change will be agreed upon with customers in the advent of such.

5.2.2. Release Schedule

The first release will take place within 2 months and with the SRS features introduced, the following updates will take each week for three weeks and the next minor within 2 months after the last fix release weeks. The fix releases will through their iterations, enhance system usability and fix any found bugs.

5.2.3. Release Impacts

In this section, the system impacts associated with each release and the processes to be changed as a result of the deployment defined in this release plan will be identified. After each launch, the development team will make changes to the system that will entail revisions to the other parts of the project, adding additional workload to the team. Therefore, any modifications in the design of the system will affect the future development of the system.

In addition, although it is the duty of each participant to accomplish their tasks the project manager will oversee the entire process. The project manager still needs to keep a close eye on the progress of the project and anticipate any potential consequences of each release. In the case of an accident, the project manager must ensure that the workload of each team member is properly balanced. In summary, the project manager should not be biased and should delegate all roles and duties equally to each member of the team.

5.2.4. Release Notification

The respective stakeholders are notified as listed below when a release version is produced. Procedures for informing the various stakeholders, the data included in the notice, the time for monitoring before publication and the information receipt will be included.

Stakeholders	Communication	Message	delivery time
Developers	Group chat, Wiki, Repo and meetings	changes in: - fixes - new functionality - UI - important implications - architecture - interface changes - api changes	in the following meeting within one week
Client	Emails and meetings	changes in: - fixes - new functionality - UI - important implications	one week after developer meeting
Users	Popup in system upon startup	changes in: - fixes - new functionality	on the day of the release

6. GLOSSARY AND ACRONYMS

Term Name	Term Definition
UCD	Use Case Diagram
SRS	System Requirement Specifications
SDLC	Software Development Life Cycle