SmartLib

RELEASE PLAN

Version *1.1*
5/4/2021

Team Xeon

Document Number: SL-10

REVISION HISTORY

Version	Date	Author	Description of Changes	Approved by
1.0	01/04/2021	Kong Hou Jing	Baseline Version	Kenny Voo Tze Rung
1.1	05/04/2021	Kong Hou Jing	Amendment to release rational	Kenny Voo Tze Rung

APPROVALS

Submitting Organization’s Approving Authority:

Signature	Printed Name	Date	Phone Number
	Kenny Voo	5th April 2021	8188 8188

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This release plan is a detailed specification of the features of SmartLib, and the upcoming release versions of their date of availability. The aim of this release plan is to aid the team with the team's progress and goals, as well as the team's priorities. With the aid of this plan, it allows the identification of criteria and goals for each release version based on the significance and priority of the task, while practicing continuous integration along the project timeline.

The intended audience for the Release Plan is for the entire project team, including the project manager, development team and QA team. The development team will be in charge of building the release package. The QA team will be responsible for the quality management of the release, and the project manager will be in charge of overviewing and the approval of the release.

2. REFERENCED DOCUMENTS

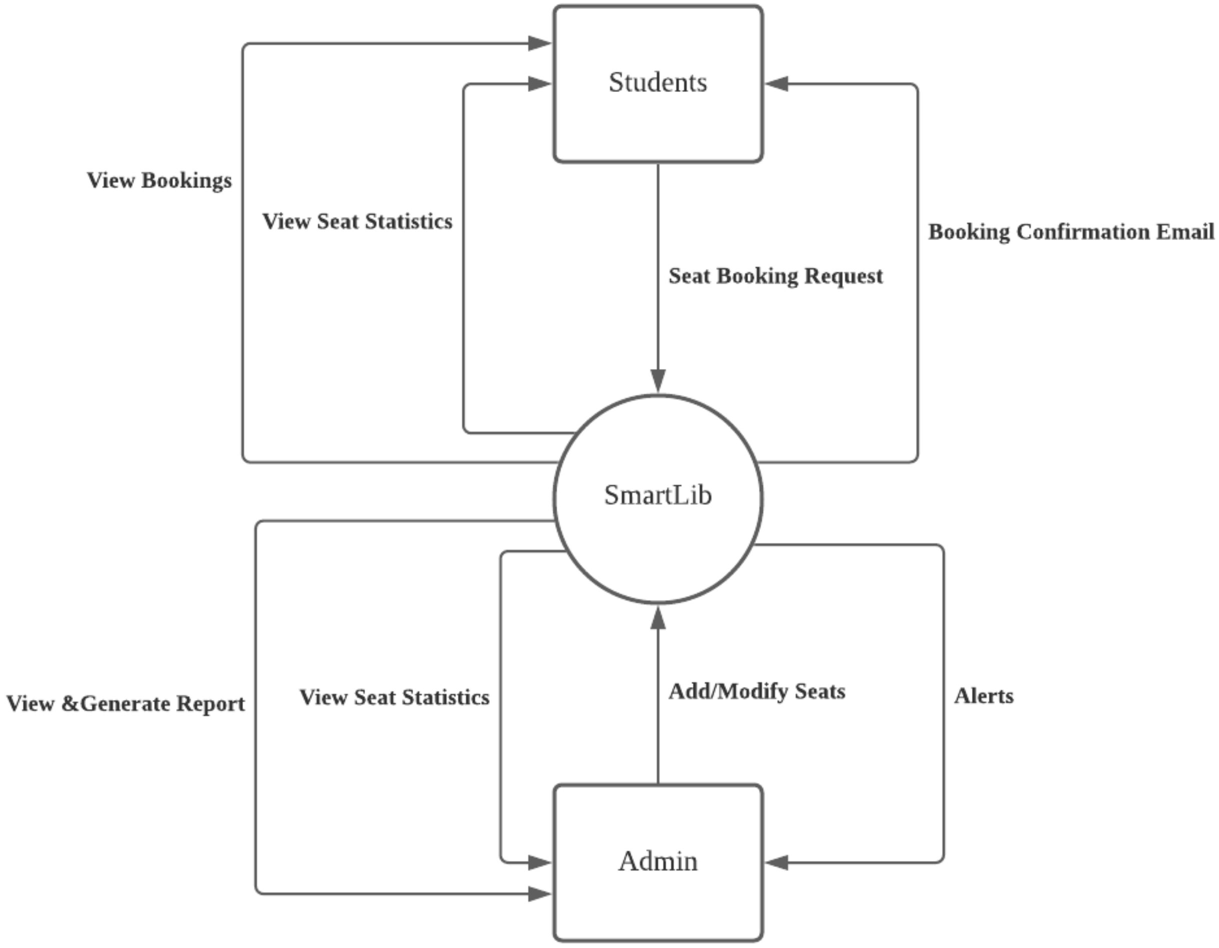
Table 1: Referenced Documents

Document Name	Document Number	Issuance Date
Project Plan	SL-05	13/2/2021
Software Requirement Specification	SL-03	5/4/2021
Risk Management Plan	SL-06	24/2/2021
Change Management Plan	SL-09	30/4/2021

3. OVERVIEW

Below shows the high-level context diagram for SmartLib, which shows the overview of the functionalities of the system.

Figure 1 : System Context Diagram



4. ASSUMPTIONS, CONSTRAINTS, RISKS

4.1. Assumptions

The team members of this project will be able to complete all the tasks assigned to them and will be committed to the project from the start to end. The cost of this project will not change during the whole duration. The development and architecture of the application will not change any point during the development.

4.2. Constraints

4.2.1. Time

The timeline for the project is 6 months, starting from Jan 2021 to June 2021. The library client wants to have the application ready before the next school semester starts. As such, the core functionality of the application must be ready before the deadline.

4.2.2. Budget

The budget for this project is limited and fixed. The application must be done within the budget allocated and using the resources available.

4.3. Risks

The table below shows the identified risks and the approach to each risk. More details to the risk management can be found in the Risk Management Plan.

Table 2: Risks and Approach

Risk Type	Risks	Approach
Technical	Requirements inflation which introduces new features which were not identified during planning phase. These new features threaten project time estimations.	**Mitigation.** **Project Manager has to communicate effectively with project sponsors as well as project team to ensure that new requirements are suitable and does not severely impact project timeline.**
	Poor quality of code	**Avoidance.**

	given resulting in higher frequency of crashes and inability to handle edge cases. Poor quality code also increases the difficulty of implementing new features.	**Experienced team members will be paired with inexperienced team members to do pair coding as well as code review. This helps to ensure that code written is of good quality.**
	Lack of end-user engagement which can advise the features that are needed or those that are redundant during software development.	**Mitigation.** **Developers have to communicate with end-users more effectively and consistently to ensure time is not wasted on redundant features.**
Management	Poor productivity which results from lack of sense of urgency to work in earnest due to long project timelines in the earlier stages.	**Mitigation.** **Project deliverables to be submitted at the end of every sprint helps to keep the project team on their toes and constantly deliver work.**
	Poor estimation of project timelines and workload due to unforeseen features and risks	**Mitigation.** **Team members have to communicate and collaborate consistently such that everyone is updated on project progress. It is important to also allow for extra time for timeline estimation during project planning to take into account unforeseen factors.**
	Health issues which may result in missing key team members during critical times, which may affect code quality and efficiency.	**Acceptance.** **It is not possible to have all team members be healthy and illnesses may arise due to global events or flu season. However, the impact can be mitigated by adopting code sharing practices and pair coding to ensure that the project does not**

		suffer derailment should a key member be unavailable.

5. RELEASE APPROACH

5.1. Rationale

As this project is based on the Agile Methodology for the SDLC approach, the project will be progressed and developed in phases incrementally, and there will be multiple releases of the SmartLib system. After the implementation and testing phases are completed from the backlog, the team will add the deployment task from the backlog. An initial release will be built and deployed (Version 1.0.0), and it will serve as a baseline release version for the SmartLib system.

After the deployment of the initial version, any reported bugs and issues will be identified. Any new system features or requirements can also be suggested. They will be reviewed by the CCB and procedures will be followed as stated in the change management plan. The changes will then be released in subsequent versions, either as a patch, a minor release or a major release.

5.2. Release Strategy

The delivery of the business product will be segmented into specific releases. The initial release (Version 1.0.0) will be released with the functionalities in the SRS fully implemented, and subsequent releases will be of bug fixes and new feature/ functionality rollouts.

The release strategy is for a phased function rollout, as the product will be released based on new features and functionalities apart from bug fixes. On the other hand, a phased user base rollout is not suitable for our product as the release needs to be deployed for all user bases (both admins and students) for the product to function as intended.

The initial release shall contain the core functionalities of SmartLib, as stated in the SRS. The next patch or minor release will contain bug fixes, as well as minor improvements to the system. The next major release will contain new major system features or system overhauls. This will occur after the issues and proposed changes have been approved by the CCB and gone through the necessary change procedures.

5.2.1. Release Content

With reference to the release strategy, there will be an initial release of the SmartLib system. These release versions will fulfil the implementation of the respective functionalities, and testing. The various release versions will acknowledge different purposes which will be stated in the release documents.

Table 3.1: Release Content - Initial Release

Release Version 1.0.0	
Functionality	**Description**
Image Detection System	Fully implemented as defined in initial SRS
Database System	Fully implemented as defined in initial SRS
Front-end Interactive Webpage: - Login and registration system - Seat reservation system - Student account features - Admin account features	Fully implemented as defined in initial SRS
Other back-end features: - Booking confirmation Email system - Data logging for report generation	Fully implemented as defined in initial SRS

Table 3.2: Release Content - Minor Release

Release Version 1.1.0 (Minor Release)	
Functionality	**Description**
Image Detection System	Bug fixes and minor improvements
Database System	Bug fixes and minor improvements
Front-end Interactive Webpage: - Login and registration system - Seat reservation system - Student account features - Admin account features	Bug fixes and minor improvements
Other back-end features: - Booking confirmation Email system - Data logging for report generation	Bug fixes and minor improvements

Table 3.3: Release Content - Major Release

Release Version 2.0.0 (Major Release)	
Functionality	**Description**
Image Detection System	Introduction of new major features or system overhaul

Database System	Introduction of new major features or system overhaul
Front-end Interactive Webpage	Introduction of new major features or system overhaul
Other back-end features	Introduction of new major features or system overhaul

5.2.2. Release Schedule

As scheduled, the first release will be expected to be deployed in the 4th month of the project timeline, with the functionalities in the SRS implemented. Subsequent minor releases will each take a minimum of 1 month to be completed, whereas major releases which include the addition of a major feature or system overhaul will take a minimum of 3 months to be completed, with the inclusion of change management procedures, and the revision of the documentations, implementation, testing and release processes.

5.2.3. Release Impacts

For each new release, the development team is required to make changes to the system, which may indirectly affect other parts of the system, thus changes might be required for other affected parts of the system. This will increase the workload for the team. Any modifications to the system functionalities might also affect future developments to the system. Thus, the project manager has to oversee the entire process, to foresee any potential risks and consequences each release might bring about.

With each new release of the system, there will also be a downtime of the system where the entire system has to be taken offline to commence the deployment of the system. During this downtime, users will not be able to access the system. This impact can be mitigated by scheduling the deployment of the release at a non-business hour of the library (e.g after midnight). In the unforeseen circumstance where the deployment has to take effect in the business hours of the library, all users will be notified via email of the system downtime and the estimated time until the system will be resumed online.

5.2.4. Release Notification

After which a release version is generated, the respective stakeholders will be notified as stated below. Methods of how the stakeholder will be notified, what information will be included in each notification and prior to release, the timeframes for receipt of the information, will all be included.

Table 4: Release Notification

Stakeholders	**Method of Providing Notification**	**Information Included in Notification**	**Timeframes for Receipt of Information**
Users	Emails, System Notifications	Change log	Upon deployment of new release
Team Members	Meetings, Emails	What changes are to be made, known issues / bugs to be fixed, and deadlines of release implementation	Immediately after the change has been confirmed by the CCB to implement the new release version
Investors & Sponsors	Meetings, Emails	What changes are to be made, known issues / bugs to be fixed.	1 week prior to version release

6. GLOSSARY

7. ACRONYMS

CCB	Change Control Board
SRS	System Requirement Specification
SDLC	Software Development Lifecycle
QA	Quality Assurance

8. APPENDICES