Team # - Project Title

Project Proposal & Plan

# Introduction (20 points)

## Project Overview and Statement of Proposal

We propose to develop a product that allows for multi-device notification integration. A similar program already exists, called PushBullet, however not only is PushBullet no longer maintained, a useable linux client does not exist. The goal of this project is to create an alternative to PushBullet, allowing users to forward notifications and view notifications on all of their devices. This is meant to contribute to the optimization of a users workflow, and provide them with an alternative when they’re not able to access all of their devices. We will achieve this by designing three separate applications. The first will be a desktop application written using C++ and QT, and we will use QT to provide cross-platform capability. Our second application will run on a server that we will host, and will interface with a database and provide routing from one accounts devices to the other appropriate devices. We have yet to decide what language to write this in, although it will not have a frontend component and will most likely be written in Python. The third application is the mobile application, and will be written in the Standard Android Studio language. Out initial goal is to provide an android application, and given enough time we also hope to write an iPhone application in Swift.

## Project Scope and Objectives

We would like to provide a fully fleshed out application for Android, Linux, MacOS, and Windows. It will provide notification forwarding from each of those devices to the others, and would also forward text message history from the mobile devices to the others. Out objectives are to provide notification forwarding from a mobile device to a desktop application, notification storage from one device to another, text message history from mobile devices to desktop, and a centralized server that would provide a framework for all of this to be executed. Other than these main objectives, we would like to provide an interface for iOS devices, as well as web client interface.

# Risk Management Strategy (50 points)

## Risk Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risks** | **Category** | **Probability** | **Impact** | **RMMM** |
| Potential for project to be too complex | PS | 10% | 3 |  |
| Possibility of hardware issues in regard to server | TE | 20% | 3 |  |
| Possibility of miscommunication in languages/frameworks | DE | 10% | 2 |  |
| Most of the Dev Team is inexperienced with a majority of the languages and frameworks | ST | 70% | 1 |  |

**Category values: Impact values:**

PS – Product Size Risk 4 – catastrophic

BU – Business Impact Risk 3 – critical

CU – Customer Relations Risk 2 – marginal

PR – Process Risk 1 – negligible

TE – Technology Risk

DE – Development Environment Risk

ST – Risk Associated with Staff Size and Experience

## Discussion of Risks to Be Managed

The first risk we listed is the potential for the project to be too complex, as it is a very wide scale project, with many small interconnected components. This is an issue with the size of our product, and has a likely 10% chance of occurring. If it occurs, the impact we believe would be critical. The next risk we listed is the possibility of hardware issues, specifically in regards to setting up our server that we will use to communicate and route between devices. This is a technology risk, and we believe it has a 20% chance of occurring. If it does happen, we believe is could be a critical issue, but we could recover from it by implementing a client-client architecture rather than a client-server architecture. Our next risk is the possibility of miscommunication in what framework or language is being used to develop each certain aspect of the project, for instance an issue could arise where we do not communicate changes from one framework to another. We feel this has a 10% chance of occurring, and we feel that the impact would be fairly marginal, as recovery from this would be simple. Our most significant risk would be that most of the dev team is unfamiliar with a majority of the frameworks being used and languages. This has a large chance of occurring, although we feel it would be a negligible issue as each of us is prepared to learn new languages and frameworks with the needed frequency.

## Risk Mitigation, Monitoring, and Management Plan

This section describes what is to be done to avoid each risk (Section 2.3.1), how the team will monitor its activities to detect when a risk becomes a problem (Section 2.3.2), and what will be done for each risk if it becomes a problem (Section 2.3.3).

### Risk Mitigation

Explain how each risk will be avoided. (Keep in mind that some activities may help to mitigate more than one risk while some risks may require multiple mitigation activities.)

| **Risk** | **How to avoid** |
| --- | --- |
| Potential for project to be too complex | Stay on top of tasks, make sure everyone keeps up with their goals, if unable to provide resources and help. Also adequate planning at the beginning stage will be help to avoid any later complexity issues. |
| Possibility of hardware issues in regard to server | Have a member of the team act as a Systems Administrator and make sure that it is maintained well and any issues are dealt with swiftly. |
| Possibility of miscommunication in languages/frameworks | Make sure to meet often and keep each other updated on all progress as often as possible. |
| Most of the Dev Team is inexperienced with a majority of the languages and frameworks | Make sure to read documentation and stay in constant communication with each other, be willing to provide assistance when necessary. |

### Risk Monitoring

If we attempt to implement too many features or go over your ability, we will make sure to discuss all plans and changes to the objectives, allowing us a platform to provide each other with more perspective and discussion.

To deal with potential hardware issues, we will have someone who will make sure to stay on top of the current server status and to mitigate any issues as they arise.

To deal with miscommunication, someone will maintain a progress chart that includes detailed descriptions of what is being used for each component.

To deal with inexperience we will make sure everyones code looks good and is well documented.

### Risk Management (Contingency Plans)

Explain what will be done for each risk if the risk becomes a problem. (Keep in mind that the same contingency plan may apply to more than one risk.)

# Schedule (20 points)

Below is a list of tasks associated with the project, a Gantt chart depicting task durations, dependencies and completion dates, and a summary of resource assignments for each task.

## Task List

1. …

## Timeline Chart

Gantt Chart (with dates)

## Resource Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **People** | **Hardware & Software** | **Special** |
|  |  |  |  |

# Project Resources (10 points)

## People

## Hardware and Software Resources

Indicate the anticipated software and hardware resources required to complete the project.

## Special Resources

List any special resources needed to complete the project.

# Appendices (5 points)

This section contains the breakdown of individual contributions and any additional information such as meeting logs that you would like to include in the project proposal and plan.