

Team # - Project Title

Project Proposal & Plan

1. Introduction (20 points)

1.1. 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 A similar program already exists, called PushBullet, however not only is PushBullet no longer maintained, a useable linux client does not exist.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. Our initial goal is to provide an android application, and given enough time we also hope to write an iPhone application in Swift.

1.2. 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. Our 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.

2. Risk Management Strategy (50 points)

2.1. 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:

PS – Product Size Risk

BU – Business Impact Risk

Impact values:

4 – catastrophic

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	

2.2. 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 it 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.

2.3. 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).

2.3.1. 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.

2.3.2. 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 everyone's code looks good and is well documented.

2.3.3. Risk Management (Contingency Plans)

If the project is too complex, we will take a vote and decide on how to simplify, if we still have issues we will consult an expert (Dr. Shin).

If we find that we have hardware issues, we will discuss the issues among ourselves, and if we still find ourselves unable to solve the issue, we will consult the NMT SysAdmin department and peers for advice.

If we have communication issues, we will have a meeting and vote to determine the solution to the problem.

When we encounter inexperience issues, we will thoroughly read documentation.

3. 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.

3.1. Task List

1. Develop a Desktop application
2. Make sure the Desktop application displays notifications accurately
3. Send notifications from the Desktop to other devices
4. Develop an android application
5. Makes sure android application can receive notification
6. Makes sure android application sends notifications efficiently
7. Develop a server side application
8. Make sure application can receive and send packets efficiently
9. Make sure application can determine source and destinations for each notification
10. Make sure application routes notifications efficiently
11. Develop a networking framework and system for inter device communication
12. (Non-Critical) Develop an iPhone application
13. (Non-Critical) Develop a web application

3.2. Timeline Chart

Table 1

	Week 1 (Feb 11- Feb 17)	Week 2 (Feb 18-24)	Week 3 (Feb 25- March 3)	Week 4 (March 4 - 10)	Week 5 (March 11 - 17)	Week 6 (March 18- 24)	Week 7 (March 25- 31)	Week 8 (April 1 - 7)	Week 9 (April 8 - 14)	Week 10 (April 15-21)	Week 11 (April 22-28)
Develop a Desktop Application											
Develop an android application											
Make sure Desktop application displays notifications accurately											
Makes sure android application sends notifications reliably											
Develop a networking framework and system for inter device communication											
Make sure application can receive and send packets efficiently											
Send notifications from the Desktop to other devices											
Develop a server side application											
Make sure application can determine source and destinations for each notification											
Make sure application routes notification efficiently											
Makes sure android application can receive notification											

3.3. Resource Table

Task	People	Hardware & Software	Special
Develop a Desktop Application	Marek Listy, Ole-Jager Hoffstuen, Jacob Garcia	C++, QT	
Make sure Desktop application displays notifications accurately	Jacob Garcia	C++	
Send notifications from Desktop to other devices	Jacob Garcia	QT	
Develop an Android application	Alexander Benson, Marek Listy	Java	
Make sure Android application can receive notification	Alexander Benson, Jacob Garcia	Java	
Make sure Android app sends notification	Alexander Benson, Marek Listy	Java	
Develop a Server Side Application	Alexander Benson, Jacob Garcia, Ole Jager Hoffstuen	Java, SQL, Derby	
Make sure Server can send and receive notifications	Alexander Benson, Jacob Garcia	Java	
Make sure Server can determine source and destinations for each notification	Alexander Benson, Jacob Garcia, Ole Jager Hoffstuen	Java, SQL, Derby	
Make sure Server routes notifications efficiently	Jacob Garcia	Java	
Develop a network framework and system	Jacob Garcia		

4. Project Resources (10 points)

4.1. People

1. Alexander Benson
2. Marek Listy
3. Ole-Jager Hoffstuen
4. Jacob Garcia

4.2. Hardware and Software Resources

We will need to use Android Studio, Eclipse, C++ THING, QT THING, and a Derby SQL server. We will host the server on an old laptop that belongs to a member of the team, and we will use that for our backend.

4.3. Special Resources

None necessary.

5. 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.