|  |
| --- |
| File Upload System |
| Project Statement |

|  |
| --- |
| Connor Oliver, Jordan Campbell,  David Campbell, Anson Carmody  1-18-2016 |

# Table of Contents

[Table of Contents 1](#_Toc440821080)

[Document Changes 2](#_Toc440821081)

[Project Summary 3](#_Toc440821082)

[Project Objective 4](#_Toc440821083)

[Project Deliverables 5](#_Toc440821084)

[Assumptions 7](#_Toc440821085)

[Constraints 8](#_Toc440821086)

[Decision Making 9](#_Toc440821087)

[Appendix A – Change Request Form 10](#_Toc440821088)

[Appendix B – Status Report Form 11](#_Toc440821089)

[Appendix C – Team Contract 12](#_Toc440821090)

[Standards and Conventions 13](#_Toc440821091)

# Document Changes

|  |  |  |
| --- | --- | --- |
| Date | Changed By | Change Description |
| 17-07-15 | Connor Oliver | Created Document, Added my sections and Jordan’s. |
|  |  |  |

# Project Summary

Project Design Team

**Project Manager:** Connor Oliver

**TBD:** Jordan Campbell

**TBD:** Anson Carmody

**TDB:** David Campbell

Project Details

**System Name:** N/A

**System Acronym:** N/A

**Start Date:** N/A

**End Date:** N/A

**Budget Estimate:** $XXXX.XX

# Project Objective

The objective of the proposed project is to provide a temporary file upload service. By allowing users to upload files temporarily through a desktop client this service would make it easy for users to share files which don’t need to be stored permanently on the internet, such as work-in-progress documents or media. This would save resources on the server and in theory provide users with a simple way of sharing files without having to manage a shared document directory or cloud service.

# Project Deliverables

Listed below are the deliverables for the project, including analysis and design documentation for use in implementing and operating the system, as well as the individual system components which will be developed.

* Management Summary
* System Overview
  + Deliverables
  + Assumptions
  + Constraints
  + Non-Deliverables
* System Key Requirements
  + Outputs
  + Inputs
  + Data
  + Process
  + Security
* System Modeling
  + System Use Case Diagram
  + System Class Diagram
  + System Event Table
  + Primary Use Case Diagrams
  + Primary Use Case Detailed Descriptions
  + Primary Use Case Activity Diagrams
  + Primary Use Case Sequence Diagrams
* System Component Details
  + Program Design
  + Output Design
  + Input Design
  + Database Design
  + Support Processing Design
* Environmental Requirements
* Implementation Requirements
* System Components
  + Database
  + File Upload and Retrieval Server
  + Web Portal
  + Client File Upload Application
* Appendices

# Assumptions

During development we are making the following assumptions about the resources available and the use of the finished system:

1. Unlimited storage capacity on server
2. Appropriate hardware running appropriate operating systems for all components will be available
3. Server will run on an optimal internet connection for file distribution
4. Client and server applications will always have an internet connection available
5. Users will not upload corrupt or malicious files

# Constraints

The system will enforce the following constraints to ensure safety of client data and the system itself:

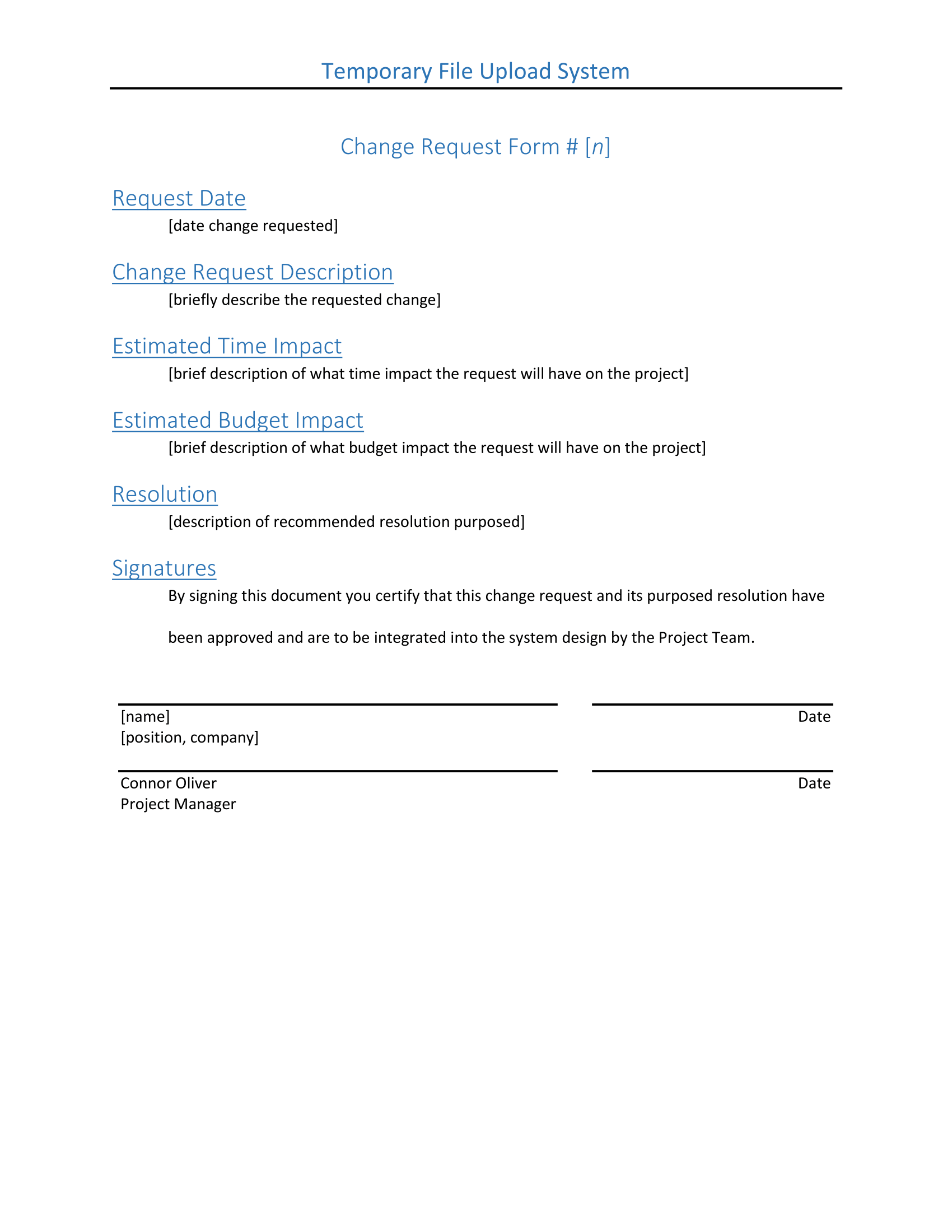
1. All user inputs such as login information and file information will be sanitized and validated
2. User login names and passwords will be encrypted before being sent to the server and retrieved
3. Fixed limit on file sizes
4. User has to be authenticated to upload files

# Decision Making

Project decisions will be made according to the following procedures:

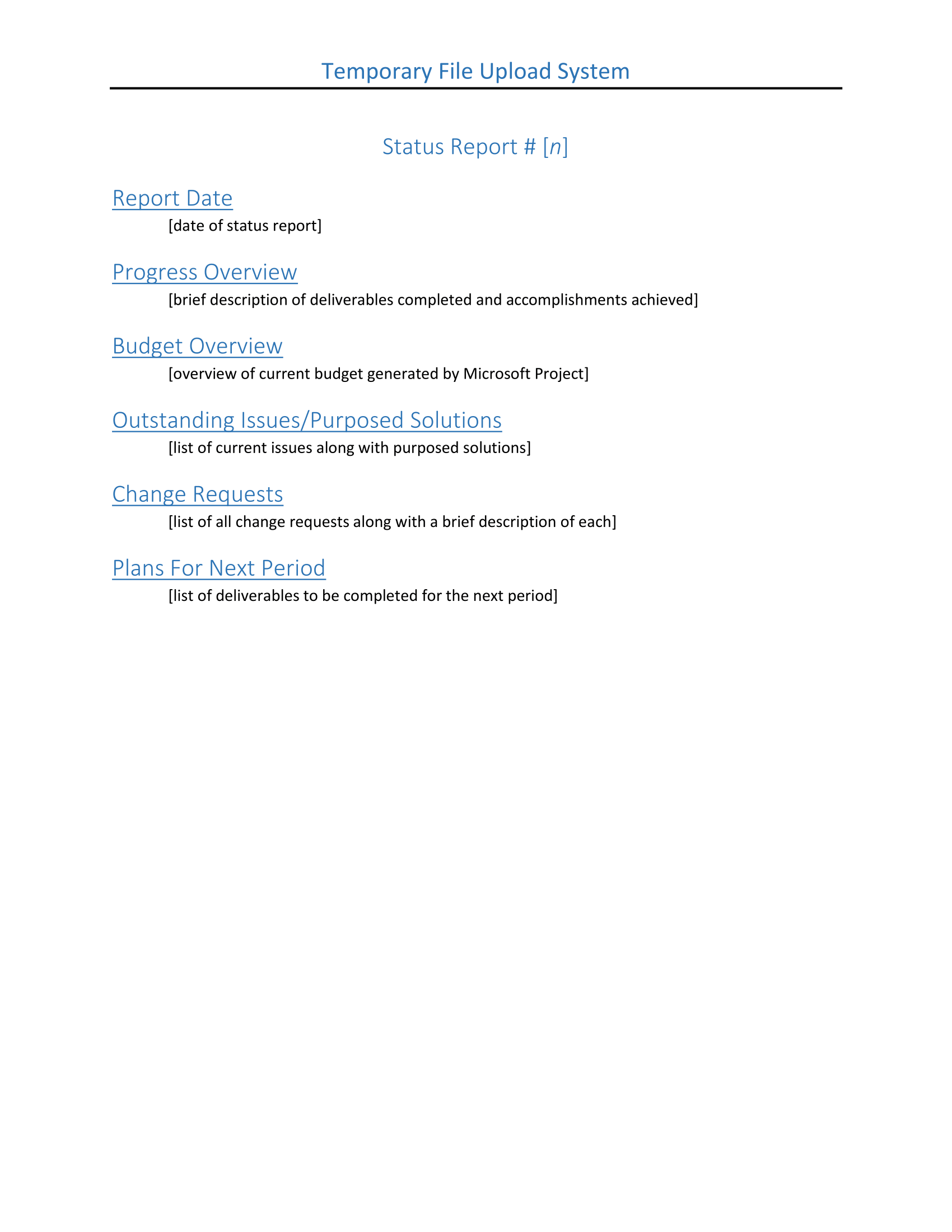
* Proposed changes regarding project business requirements will be discussed between the Project Manager and the Client, with the Client having final say.
* Proposed changes which may impact the budget will first be discussed between the members of the design team to see if any alternatives may be available.
* If a proposed change will have a budgetary impact it will be brought to the Client for final decision.
* Proposed changes without an impact on the budget will be discussed between team members, with the Project Manager having final say.

# Appendix A – Change Request Form





# Appendix B – Status Report Form





# Appendix C – Team Contract

Content

# Standards and Conventions

Storage Units

During development the project will be stored on Git, with local copies of the repository on each team member’s development computers.

When running in production the server side code will being running on some form of server and the client level programs will be downloaded from the server and run on client’s personal machines.

Executable and Source Code File Naming

Source code file names should be relevant to the files purpose and should follow the naming format guidelines of the language being used. If no naming guideline exists for the selected language files will be named using Pascal Case.

Executable files should be named something relevant to what they do using Pascal Case. If the executable is for a compiled version of a program it should be given that programs name and contain a space between any words in it.

Source Code Header Documentation

All source code files should have a header that contains the name of the file, the creator’s name, date created, date last edited and brief description of the file and its purpose.

The following is an example of a source code header:

/\*

\* date.c

\* Author: Yuri Gagarin

\* Created: Dec 21, 1991

\* Last Edited: Jan 29, 1993

\* This file is used to process and format dates within the program.

\*/

Source Code Version Control

Source code will be controlled through the Git version control system. Each commit message will be a brief description of the changes made to the source.

The master branch should always contain a working copy of the program(s), while code being actively edited should be in separate branches until completed.

Database and Table Naming

Naming for databases should following the following naming conventions:

* Database Names: Pascal Case
* Table Names: Camel Case
* Column Names: Snake Case

Variable, Function and Class Naming

Variable, function and class names should follow the general style guidelines set forth by the language being used. For example when using Python one should use PEP8, or when using C# one should use Microsoft’s style guidelines.

If no common guideline exists for a language, the team will meet before doing any work in that language and document their own guidelines to follow.

Document Naming

Documents should be named the same as the document title, or in cases without a title something relevant to the contents of the document. Names should be written using capital letters on words with spaces between each word in their name *(ex. “A Long File.txt”)*.