Software Requirements Specification

for

Version 1.0 approved

Prepared by Kelsey Cameron,

Alex Clavelle,

Taylor Lapeyre,

John Anny,

Andrew Bergeron

Forty Three Thirty LLC

October 6, 2014

Table of Contents

Table of Contents

Revision History

1. Introduction
- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Product Scope
- 1.5 References

2. Overall Description
- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 User Documentation
- 2.7 Assumptions and Dependencies

3. External Interface Requirements
- 3.1 User Interfaces
- 3.2 Hardware Interfaces
- 3.3 Software Interfaces
- 3.4 Communications Interfaces

4. System Features
- 4.1 System Feature 1
- 4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements
- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes
- 5.5 Business Rules

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a thorough description of the web application, WishWheel. This document will specify WishWheel's purpose, features, target users, hardware and software requirements, and user interfaces. The complete system of version 1.0 will be covered within this SRS.

1.2 Document Conventions

This document is broken into sections then sub sections. Section titles are bold, eighteen point, plum color, and Times New Roman. Sub section titles are bold, fourteen point, plum color, and Times New Roman. The paragraph text is eleven point, black, Times New Roman.

1.3 Intended Audience and Reading Suggestions

This document is intended for all the WishWheel stakeholders. Those who are interested in an overview of the project should continue to read section one, section two, and may wish to reference section six.
Readers who are interested in more details such as the features of the project should read on to section three. Section four offers further technical details for those who wish it. Those who would like a preview of the user interface should also see section four.
Readers interested in nontechnical features and requirement of WishWheel should read section five. Section five covers performance, safety, security, and quality attributes.

1.4 Product Scope

The goal of WishWheel is to facilitate a clear and simple communication between a gift receiver and a gift giver. During the holidays it can be confusing who is purchasing what and for whom. WishWheel allows users to post wish list, reserve an item off another's list, and set up events with groups. WishWheel is a free to use web application.

1.5 References

This document features some terminology which readers may be unfamiliar with. See Appendix A for a list of these terms and their definitions.

2.

Overall Description

2.1 Product Perspective

WishWheel is a new self-contained system. The system will be a single page web application which will be used to create a clear understanding among families and friends, at least where gift giving is concerned.
<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

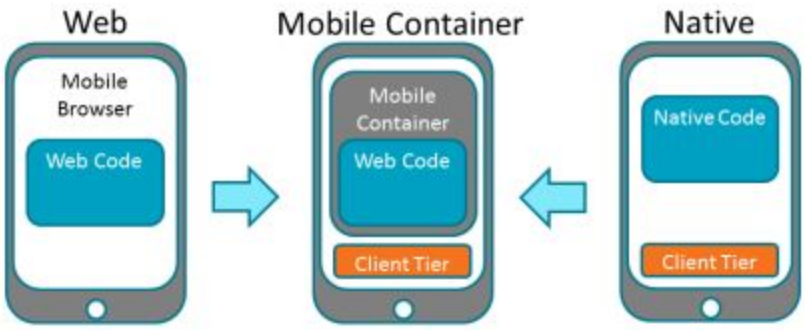
2.2 Product Functions

WishWheel allows a user to create a wish list. The user can view another's wish list and reserve a gift on that wish list which the user wishes to purchase. All the users, excluding creator of the wish list, can see the reserved gift. The ability to see who is buying what and for whom clears up the possibilities of someone receiving two of the same gift or not receiving anything at all.
WishWheel gives a user the ability to create an event. An event can be anything from Christmas to a family get-together just because. The user gives the event a title, guest list, and their wish list. The users on the guest list can then join the event and create their wish lists.
WishWheel users can create reminders. This reminder will be emailed to the user for the time the user schedules. Reminders can remind the user of whatever they set it to.

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment



http://blog.equinox.co.nz/blog/Lists/Photos/MobileApplicationDevelopmentArchitectures.png

The WishWheel software will act as a mobile application for iPhone and Android iOS 8, KitKat 4.4.

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

network performance:
-transfer rates at 4G vs transfer rates at wifi
(lets say you go to view a wishwheel profile and they have 1000 gifts for 1 wheel. The developer would not want to populate the page with EVERY single gift in one request. obviously use a paging technique.)
The developer should not request more than a limited # of gifts from hosted database at one time for performance reasons.

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices.

Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

no one is getting hurt by making wishes other than their wishes not coming true

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

Fraud/Identity Theft. All Wish-Wheel users will have a profile with some personal information.
refer to section4.1.1???
A minimal amount of information about a user will be visible to other users. username, profile picture, and their shared wishwheels with gifts listed.

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

-efficient
discuss # of clicks or keystrokes required to perform certain tasks.
design strategy of a single page application leads to efficiency

-effective
usability relies on user feedback through evaluation of the product design. the evaluation will involve watching real people use the prototype and using what is learned from the users experience.

-engaging
The interface will be user-centered, engaging, and satisfying to use. The style of the visual presentation will be simple to read and navigate. graphic images and personal color scheme will be available to users.

-error tolerance
the interface will make it difficult to take incorrect actions that could lead to errors. we will design links and buttons to very distinctive so that the user can make clear choices.

-easy to learn
interface allows users to build their knowledge without deliberate effort. we will include built-in instructions for general task. ie a big green arrow pointing at a "+" sign for creating a wish wheel. Or a gift wrapped in colorful paper that is transitioning between 2 different states that easily catch the users eye.

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

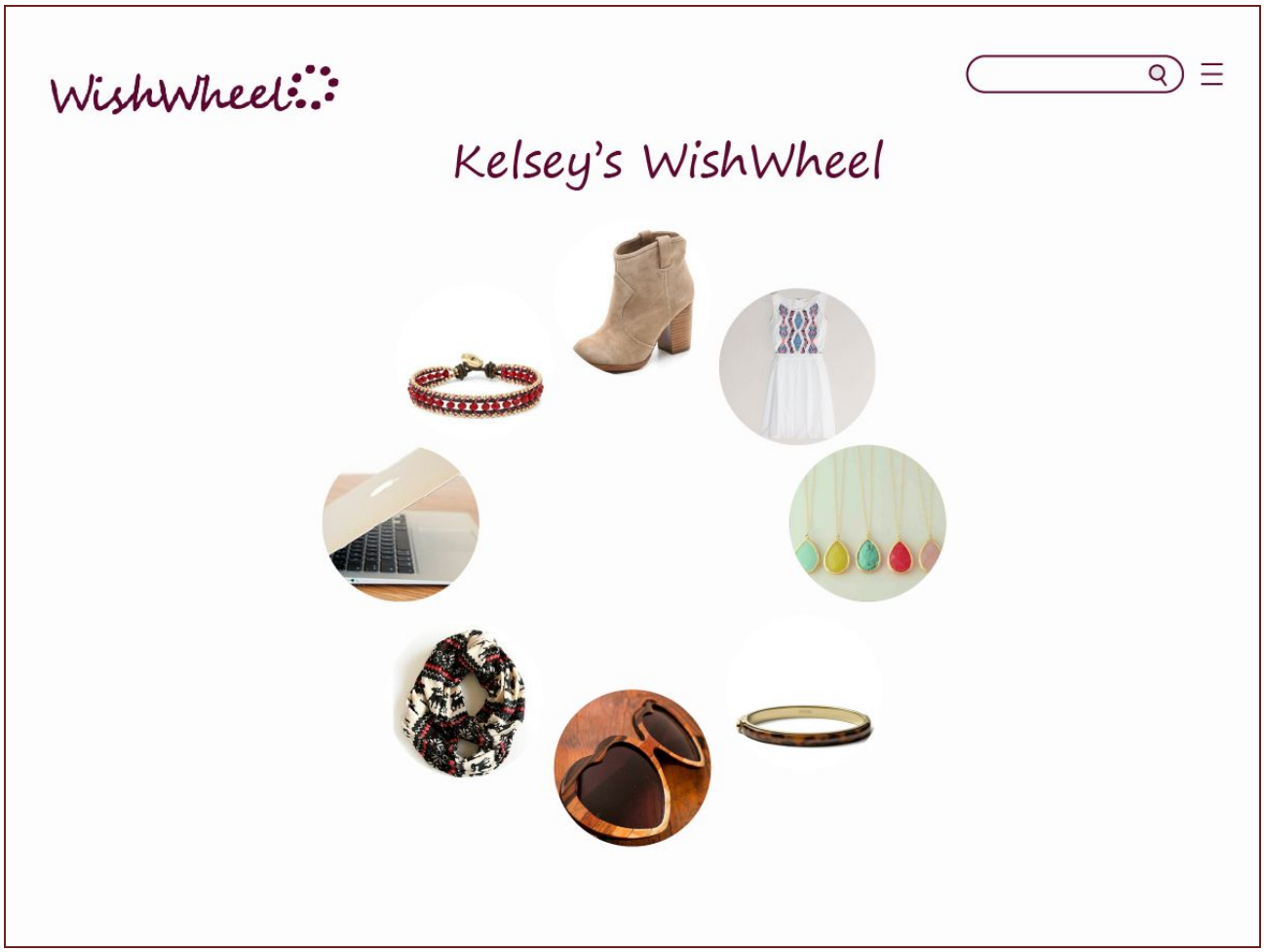
<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>


WishWheel
Kelsey's WishWheel