

IWISH DOCUMENTATION

CONTENTS

1. Brief Description of iWish	02
2. Process Followed by Team	03
3. Requirements & Specifications	04
4. Architecture & Design	05
5. Future Plans	08

BRIEF DESCRIPTION OF IWISH

iWish is a generic iPhone wish list app that is essentially a gift register integrated with an email alert system, gift pooling system and a social network. iWish is an app that allows you to register the gifts that you want when you can think of it - you can add as much detail as you want, including the price, where to get it or even an Amazon link to the product. The app allows your friends to see those gifts and know what to buy you and procure it when the date approaches.

PROCESS FOLLOWED BY TEAM

iWish was built using the Extreme Programming (XP) paradigm. Some of the core tenets of XP are creating user stories to guide the development of use cases, dividing these user stories amongst short iterations which constitute releases, thoroughly unit-testing code, recurrent refactoring and having a close relationship between team members and the customer.

For the entirety of the project we were mostly able to satisfy all but one of these XP principals. Testing proved to be problematic for the development team. Attempts were made to integrate multiple frameworks, namely KIF and Quick, for user interface testing but none proved to be beneficial. The Quick framework could not be compiled with our project and by the time we got the KIF framework to work, time constraints demanded our efforts to be focused elsewhere.

We did have a little trouble with iterative development at a few points from version control problems losing code and difficult use cases taking more than one iteration but on the whole this part of XP design went smoothly.

Refactoring and collaborative development are two parts of the XP process we handle well. Refactoring was done frequently during the design process while developing new user stories as well as at the end of the process when preparing to package the final build. Communication between our team members was largely done at weekly group meetings but was also done during pair programming sessions and through email correspondents. At no point was there any disconnect between developers as we collaborated often and speedily on issues. Since in many ways the customer for iWish was the development team itself, communicating with the customer was also done smoothly.

REQUIREMENTS & SPECIFICATIONS

IMPLEMENTED USE CASES

- User can add an item to an iWish list
- User can remove an item from an iWish list
- User can create an event
- User can delete (cancel) an event
- User can create an account
- User can add/remove a friend
- User can search for a friend
- User can set the gift pooling option
- User can buy/pool gifts in an event
- User can receive email notifications for pooling gifts
- User can receive email notifications for account registration
- User can add/remove gift to/from and event
- User can invite friends to event
- User can accept/decline events requests
- User can edit an event
- User can edit gifts
- User can receive email notifications for event cancellations
- User can view profile page/friend's profile page
- User can access setting page

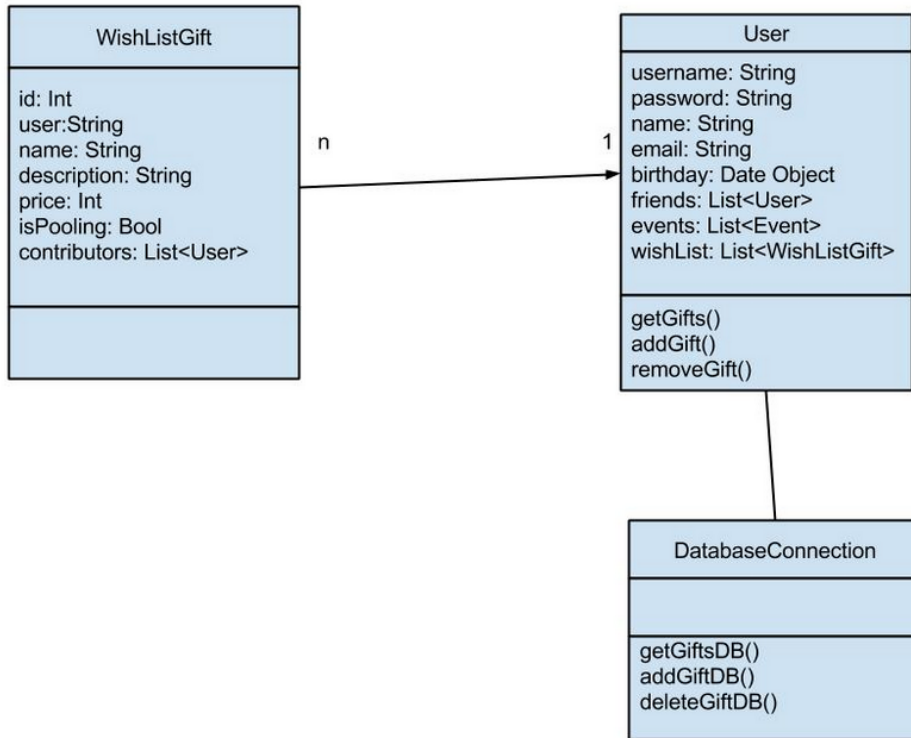
ARCHITECTURE & DESIGN

iWish was implemented using the Model-View-Controller (MVC) design pattern. This choice is due to the fact that MVC is what Apple encourages all IOS developers to use. They have designed XCode in such a way that it reinforces the MVC pattern. Every view that is shown to the user is controlled by a dedicated view controller. The models are our supporting classes such as the UserEvent class and User class.

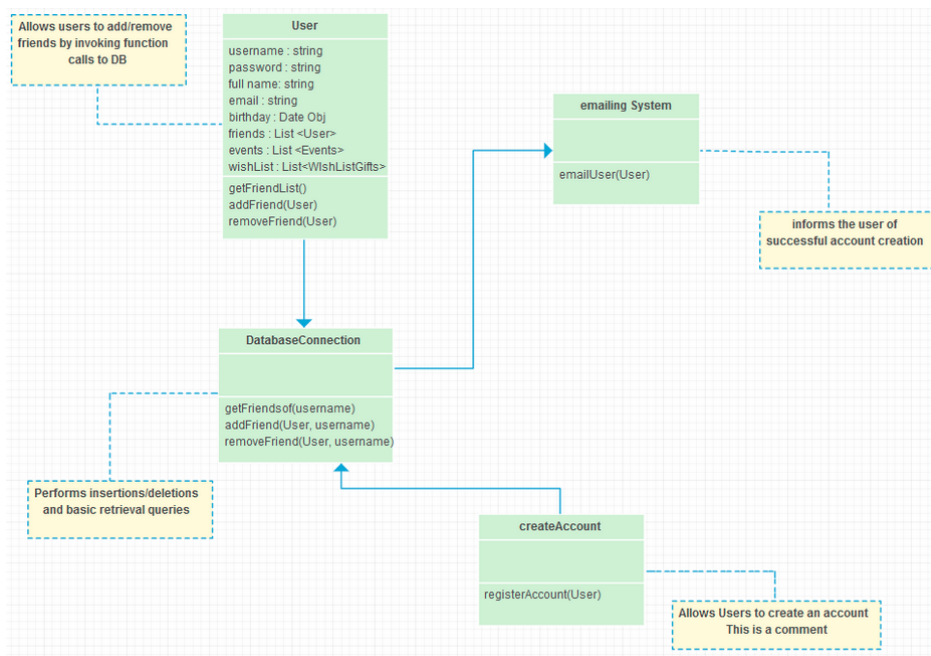
The app is broken into 5 key areas which are represented by 5 tabs in the user interface. These areas are Events, Wish List, Friends, Settings and Profile. Each area corresponds to a set of tasks that can be done by the user under their respective domain. For example, in the Friends area, the design and logic for viewing friends, searching for friends, sending friend requests and receiving friend requests can all be found. There is also a component of our project which facilitates communication between all of these 5 components and the database. Testing is implemented in a different target and therefore not a part of any releases.

UML DIAGRAMS FOR REPRESENTATIVE PARTS

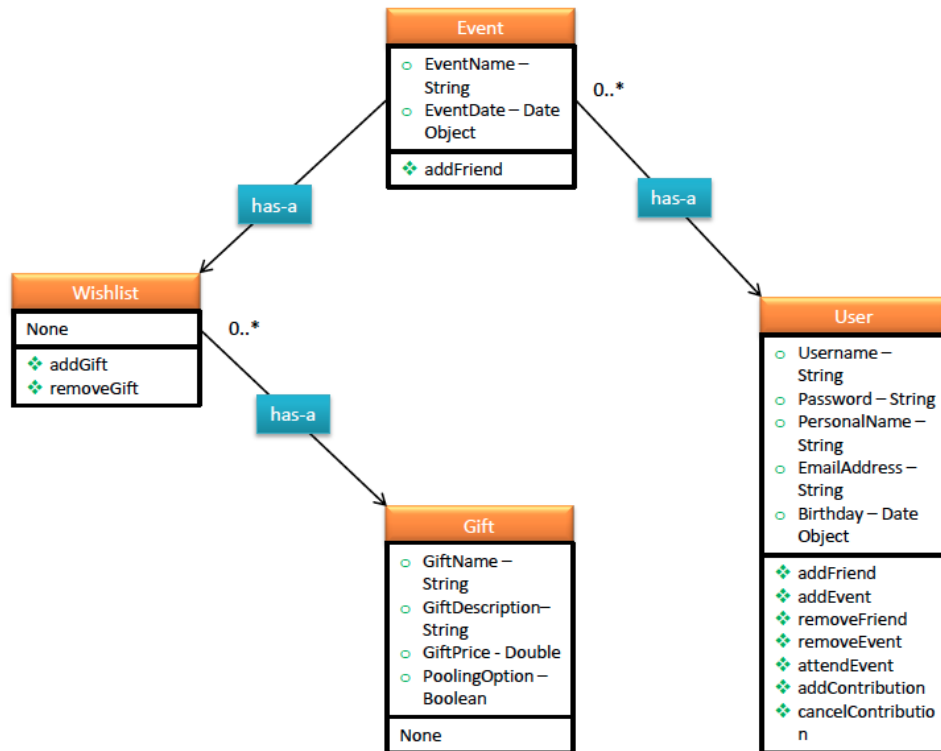
CREATE A GIFT ENTRY IN THE USER WISHLIST TABLE



CREATE A NEW USER ACCOUNT



CREATE AN EVENT AND INVITE FRIENDS



FUTURE PLANS

We do not have any future plans to extend iWish at this point. We are happy with what we currently have and think it is a complete project. There are other features that would could add but they are not core to what we set out to accomplish. We may submit the app to the app store and readjust our position with future user feedback.

If we were to extend iWish, however, we would first adapt our current project to use CocoaPods, a library management system for IOS development. CocoaPods takes care of updating and adding libraries for us which would reduce a lot of our frustrations further down the road and be really useful with our currently unstable mailcore library.

PERSONAL REFLECTIONS

KEVIN FRENCH

I had a lot of fun working on the iWish project with my teammates and think we created a product to be proud of. The design process was always well coordinated, communicating plans and problems was always beneficial and always went smoothly. The biggest things I learned was really more of a reinforcement of the idea that planning and organization is pivotal. The problems we had with testing could have probably been solved had we taken better care to plan for unpredictable issues. Similarly, the troubles we faced with version control could have probably been solved had we slowed down and addressed them when they came up instead of pushing them until later.

ZACHARY BOHLIN

I'm happy with the project choice I made. The first reason is the language and software development process we used on the project. I enjoyed learning Apples new Swift language. I had been trying to motivate myself

to start learning it, and this project was the perfect motivation for that. Also, getting to use the XP software process helped me to gain experience in it for future projects that I work on. The second reason I'm glad with the project was the fact that the concept of the project was interesting to me. The idea was novel and seemed to me like it would be fun to implement. The last reason I'm glad with the project was the team I worked with. They were all hard workers and were able to finish all that was required of them. If I had another project class to take here, I would definitely choose this same team.

OLEKSIY KAMENYEV

I enjoyed working with my team on the iWish project, and I am happy with the process we followed and the product we created. Members of the team was always willing to communicate with one another, and we were able to maintain a very organized workflow, despite the problems we had with developing in a new language and new development environment. I learned that keeping an organized software development process such as XP is crucial to obtaining a finished product, as it allowed us to be able to deal with many of the problems we had efficiently. We were able to merge successfully because we were able to organize our merging, and when I did not own a Macbook, through pair programming and teamwork, I was able to contribute to the project. We still could not solve the problem with automated test cases, as Swift was a new language to the majority of the team, and, possibly if we had more time, we could have solved the problems with testing through better organization. Overall, I think the project turned out to be a success.

Swapnil Shah

iWish was an exciting project for me, since it was the first time I ever worked on iOS development. Learning a new programming language

“Swift” was also quite challenging. All the members of the team were very friendly, and enthusiastic about the project. Communication was never an issue throughout the project. All the workload throughout the iterations was organized well, and we managed to finish all the requirements on time. Pair programming was the most important aspect for me since I didn’t have a macbook, however it never became an issue, and made the entire development process more collaborative. I learned how important different aspects of XP like pair programming and planning game can be when it comes to develop a new project from scratch. There were a few issues with merging at first, however we figured that out over time. Overall, it was a great learning experience.

Ai Vong

Working on iWish with a team demystified iOS development for me as I always thought it to be a complicated process. Now, however, the MVC pattern is very clear to me and I am aware of all the tools required for iOS development. One major issue at the start was that only three members had macbooks to develop on. Over the next few weeks, however, we all had a proper development environment which was a relief. Merging was also an issue, but our team managed to communicate well and overcome merging obstacles. In terms of process, I was able to do a lot of pair programming with my partner because they did not have a macbook. The results were pretty good as we were able to collaborate well with one another. In conclusion, I am very happy we managed to finish all of our use cases. I was able to work with great people and learn a lot from them. I will definitely take what I have learned from working on iWish and apply the lessons to a future project.

Pattanee Chutipongpattanakul

This class, this project, and this team has been an invaluable experience for me. Developing iWish was a unique experience because this is the

first time I have the opportunity of developing a project from ground up and working together with a team of this size. I have always been used to programming in smaller units, so this project enabled me to learn how crucial communication is. Working with everyone is enjoyable, seeing what my peers can bring to the table and solve problems together is truly a privilege. Other than the experience with my team, this project has also brought the opportunity to experience XP to the truest sense. Due to the lack of resources in the first few weeks of development, my sub team has come to know how to effectively pair program. I can safely say that everyone took away useful skills from this project, be it manual merging, using the same coding standards, collaboration and even a new language - Swift. We have learnt how to evaluate our project risks and make adjustments to our plan as we go. We have learnt how to organize the iterations to ensure a good workflow and proportionate distribution of use cases. I would say that, even if there were hurdles that we could not overcome, such as automatic testing, this project was a success due to the new skill set everyone has obtained from it.