Team 5 - Product Backlog

Andrew Bass, Will Borland, Pravin Sivabalan, Devin Sova

Problem Statement

The chemistry behind drugs that people rely on is complicated, and the interactions of medications is difficult to understand without consulting professionals or reading through verbose documentation. Our application allows an easy way to manage their medication on their own, without missing out on vital safety information. Marigold empowers people to be more informed consumers when it comes to medication.

Background Information

Audience

Our target user base includes all individuals who have to take medications, especially on a long term basis. Those who have additional constraints on their medicines, such as allergies or work restrictions, will especially benefit from the insights Marigold provides.

Similar Platforms

There are other existing medication management apps such as CareZone or Drug Barcode Scanner. These two apps allow users to manage their medication in seperate ways. CareZone allows the user to list their medications and reorder them. Drug Barcode Scanner allows the user to scan their prescriptions and get the information about it.

Limitations

Current solutions are effective at what they do, but they fall short in helping the user understand their current medications. These apps only allow you to track which medications you take, along with when to take them. With MariGold we want to let users understand how their medications work together, get reminders to take their medication and help people with allergies or restricted substances not take harmfull medication. We offer deeper analysis of user's medications than current solutions.

Functional Requirements

1. As a user, I would like to be able to register for a MariGold account.
2. As a user, I would like to be able to login and manage my MariGold account.
3. As a user, I would like to be able to delete my MariGold account.
4. As a user, I would like to be able to reset my password for my MariGold account.
5. As a user, I would like my information to be secure.
6. As a user, I would like to be able to manually add a medication to my medication list.
7. As a user, I would like to have MariGold scan in my medication via a photo of its label to my medication list.
8. As a user, I would like to see an overview of all of the medications I am taking.
9. As a user, I would like to be able to see detailed information about a particular medication.
10. As a user, I would like to be able to add a temporary medication to my medication list.
11. As a user, I would like to be able to edit my medication list.
12. As a user, I would like to edit the information about a specific medication.
13. As a user, I would like to be notified when to take a medication from the medication list.
14. As a user, I would like to be able to decide on what type of reminders I receive.
15. As a user, I would like to see when to take my medications in a calendar view.
16. As a user, I would like to be informed when my medications conflict.
17. As a user, I would like to be able to lookup information about medications.
18. As a user, I would like to add medication allergies to my account.

19. As a user, I would like to select what sports leagues I participate in.
20. As a user, I would like to be notified if the medication I'm taking are under the pretenses of the athletic organization I am a part of.
21. As a user, if I'm experiencing a medical symptom, I would like to have MariGold inform me if that symptom could be a side effect of one of my medications.
22. As a user, if I'm experiencing a medical symptom, I'd like to look up over-the-counter or prescription medications that could potentially help me.
23. As a user, I would like to be able to register my pharmacy so that I can have easy access to their information, for tasks like refilling.
24. As a user, I would like to have refill reminders.
25. If time allows, as a user, I would like to manage my account on a web browser.
26. If time allows, as a user, I would like to see my medications on a web browser.
27. If time allows, as a user, I would like to use Marigold from a wearable device.
28. If time allows, as a user, I would like to sign into Marigold with Google and Facebook.
29. If time allows, as a user, I would like to unlock the Marigold app with touch ID.

Non-Functional Requirements

Architecture and Performance

We plan on developing MeriGold with two seperate system, an iOS app and linux server. This will make working together easier and more efficient. The linux server will be hosted on AWS and built in Python using the Flask framework. Flask has many security features that we will implement to keep our users data safe and secure. The linux server will also host our SQL database where most of our data will be kept. The iOS application will be developed in Swift, and will use our custom API to send and receive data from the linux server. MariGold should respond in 750 ms or less to prevent user unstatafaction.

Security

Security is essential for MariGold because it will hold sensitive user information. By using the Flask framework we have access to security features such as token based authentication, password hashing and session based authentication. Flask also supports external libraries to add increased security.

Usability

The user interface needs to be simple so everyone can use it easily. Reminding users when to take their medication is a key part of our application. Local notifications will be the main reminder system. If time allows we would like the user to have the extra choice of SMS and Email.

Hosting, Deployment, and Scalability

With using a seperate front and back end we will be able to update and deploy each one separately. Our Linux server will be hosted on an AWS EC2 instance that will be attached to a Load Balancer which will prevent unscheduled downtime. We will make our API stateless, therefore allowing us to scale by creating multiple instances of our backend.

Our backend can handle a large number of concurrent requests due to our usage of AWS's Load Balancing features. The negative effects of this come from the variable cost associated with the elastic scaling of the servers. Therefore, we will very likely limit our number of users to 5,000.