The General Services Administration (GSA)

Federal Acquisition Service, Integrated Technology Service

National IT Commodity Program

4QTFHS150004

July 7, 2015

|  |  |
| --- | --- |
| ***Submitted to:*** | ***Submitted by:*** |
| Hassan Harris  Senior Contracting Officer  National IT Commodity Program  820 W. Peachtree Street Suite 820  Atlanta, GA 30308  Phone: (404) 215-8705  Email: hassan.harris@gsa.gov | Description: Description: clearavenue2 |
| Point of contact: | clearAvenue, LLC  Dr. Srini Kankanahalli  Tel: (410) 271-4171  7125, Thomas Edison Drive  Columbia, MD 21046 |
| Type of business: | Woman-Owned Small Business, 8(a) Company |
| Tax ID Number: | 611430076 |
| GSA Schedule | GS-35F-0426T |
| DUNS #: | 124289708 |
| CMMI Level: | CMMI Level 3  ISO 9001:2008 (Quality Management), ISO 20000-1:2011 (Service Delivery), and ISO 27001:2005 (Security) |

Agile Development Approach for myMedications, FDA Drug Interaction (FDADI) Application

**Table of Contents**

[1.0 clearAvenue Development Process used for the myMedications Application 3](#_Toc423979159)

[1.1 The Product Owner 3](#_Toc423979160)

[1.2Roles and Responsibilities 4](#_Toc423979161)

[2.0 Requirements Elicitation using Human Centered Design Principles 5](#_Toc423979162)

[2.1 Development Process 8](#_Toc423979163)

[2.2 Monitoring Progress 9](#_Toc423979164)

[3.0 Tools Used 10](#_Toc423979165)

[4.0 Continuous Integration, Deployment and DevOps 11](#_Toc423979166)

[5.0 Test Automation 12](#_Toc423979167)

[6.0 Continuous Deployment 13](#_Toc423979168)

[7.0 Continuous Monitoring 15](#_Toc423979169)

[8.0 User Acceptance Testing (UAT) 17](#_Toc423979170)

[8.1 Testing on Mobile Devices 17](#_Toc423979171)

[Appendix 1- JIRA Export of the User Stories, Tasks and Chores 24](#_Toc423979172)

***Revision History***

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 6/18/15 | 1.0 | Initial Version | Nelofur Damiani |
| 6/20/15 | 1.1 | Updated | Nelofur Damiani |
| 6/23/15 | 1.2 | Updated | Nelofur Damiani |
| 6/30/15 | 1.3 | Updated Language/Formatting | Christina Paleczka |
| 7/6/15 | 1.4 | Final Review | Christina Caleczka |

# 1.0 clearAvenue Development Process used for the myMedications Application

clearAvenue has utilized Scrum and Test Driven Development (TDD) based agile development methodologies and processes to create this myMedications application. Scrum requires commitment from all stakeholders. The development team worked closely with the Product Owner to create a backlog, prioritize features, and review developed functionality at the end of each Sprint. Daily scrum describes the current development activity for the Product Owner and any additional stakeholders in attendance. Sprints consist of the Sprint Planning Meeting, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective Key components of our approach is illustrated in Exhibit 1.0-1

|  |  |
| --- | --- |
| Daily Scrum (15 minutes each morning) | Development Team answers the questions:   * What did we do yesterday? * What are we doing today? * Do we have any impediments or obstacles? |
| Backlog Grooming Session (mid- way through the Sprint) | Team conducts an informal meeting which can lead to any of the following:   * Remove user stories that no longer appear relevant * Create new user stories in response to newly discovered needs * Re-assess the relative priority of stories (in light of possible new information) * Split user stories which are high priority but may be too complex for completion in coming iteration |
| Sprint Planning (Day Zero of Sprint) | Development Team discusses goals and objectives for the Sprint with the Product Owner and any other stakeholders |
| Sprint Review | Development Team demonstrates completed functionality for the Sprint to the Product Owner and any other stakeholders |
| Sprint Retrospective | Development Team discusses the closed Sprint:   * What went well? * What could be improved for the next Sprint? |

**Exhibit 1.0-1: The Scrum based Iterative development process used for the design, development and deployment of the myMedications Application**

1.1 The Product Owner

The Product Owner is responsible for maximizing the value of the product and the work of the development team and is the sole person responsible for managing the ***Product Backlog***. The product owner develops Product Backlog items known as ***User*** ***Stories***, prioritizes them to best achieve business goals and missions, and ensures

* Value of work performed by the development team,
* Product Backlog visibility and transparency of what’s next in the Product Backlog, and
* The development team understands User Stories in the Product Backlog at the appropriate level of detail.

Please see Exhibit 1.0-1 and Exhibit 1.0-2 below for the Agile/Scrum Core Process and tools used for the construction of myMedications. **For the Agile BPA myMedications application, we developed 7 sprints of 1 day duration each.**

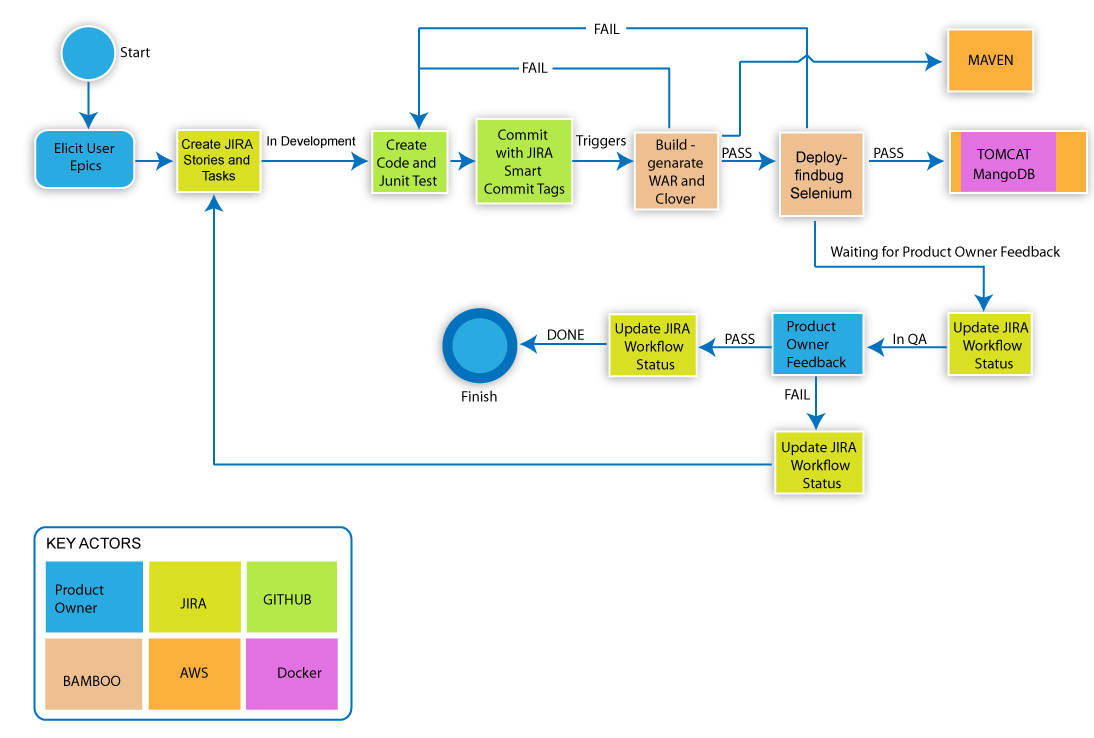


Exhibit 1.0-2 – Agile/Scrum Core Process as well as the various tools and technologies used in the construction of myMedications.

1.2Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| BPA Labor Category | Person Assigned | Responsibilities |
| 1. Product Manager (Product Owner) | Dr. Srini Kankanahalli | The Product manager/product owner is responsible for the features, quality and final signoffs on all the artifacts. |
| 2. Technical Architect | Mr. Gopal Kankanahalli | The Technical Architect is responsible for the architecture and the selection of tools and technologies and the completeness of the design. |
| 3. Interaction Designer/User Researcher/Usability Tester | Ms. Christina Paleczka | The Interaction Designer/Usability Tester is responsible for User requirements and usability testing. |
| 4. Frontend Developer | Mr. Timothy Scott | The Front End Developer is responsible for developing the Frontend of the application. |
| 5. Backend Developer | Mr. William Hunt | The Backend Developer is responsible for web development using open source technology. |
| 6. DevOps Engineer | Mr. Jeff Heath | The DevOps Engineer is responsible for engineering the application. |
| 7. Security Engineer | Mr. Jeff Heath | The Security Engineer is responsible for testing and mitigating security vulnerabilities. |
| 8. Delivery Manager | Ms. Nelofur Damiani | The Delivery Manager is responsible for managing the agile team and delivering the application. |

Exhibit 1.2-1: Roles and Responsibilities

# 2.0 Requirements Elicitation using Human Centered Design Principles

clearAvenue sent a team to a nearby retirement facility to elicit requirements to build an application to address this problem: ***As a user, I want to research my medications and be alerted through a prompt about any drug that has a recall or has interactions between my existing medications and be able to add or remove them.*** We interviewed prospective users who became stakeholders through the lifecycle. We gave them a presentation about what we do and asked for their participation in designing the interactions in a consistent and a coherent manner from their perspectives. As part of the process, we employed the following human centric design tools:

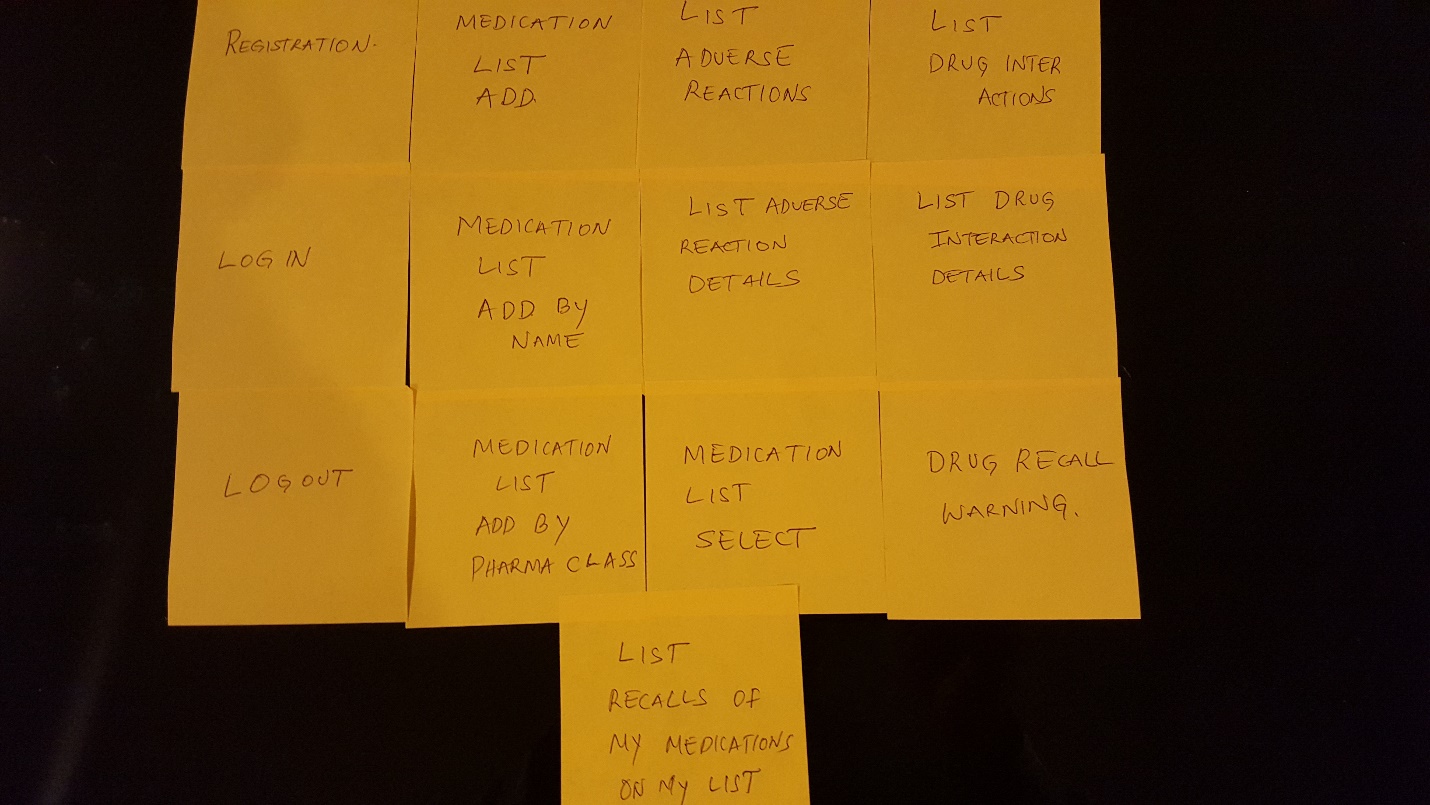


Exhibit 2.0-1: The Card Sort process to obtain user input on the features and user interaction scenarios.

1. Conducted a card sort after initial solicitation at the retirement facility with the Team. Card sorting is a brainstorming technique that is often used with non-technical stakeholders. clearAvenue worked with the group to identify basic themes and translated these ideas onto Post-it notes as a visual aid as shown above in Exhibit 2.0-1. The stakeholders arranged the Post-it notes into groups they were most familiar. From this point, the team was able to identify themes and priorities from the stakeholders. Some of the questions asked by the team to elicit themes included the following:
   * What was the user’s main concern and need for the application and its important features?
     + Drug interactions [#1 most votes]
     + Medical Device recalls [#2]
     + Food recalls [#3]
   * How would you like to find out about drug interactions for medications you are taking?
     + Anonymously with no profile [#2]
     + Anonymously but with a profile to remember their drugs but no verified registration info including email [#1]
     + Verified personal information with contact info and profile [#3]
   * Why are you concerned about your medications?
     + I take a lot of different medications and sometime forget what they all are [#2]
     + Trust but verify pharmacist and physician about adverse interaction [#1]
     + Figure out if something was recalled before I take it [#3]
2. Developed the initial set of user stories based off of these requirements gathering meetings
3. Developed Wireframes to demonstrate the ‘look and feel’ as well as desired user functionality
4. Requested volunteers to review the prototype using Google Docs on overall usability of the User Interface (UI) screens, the URL provided to the volunteer is below: (<https://docs.google.com/forms/d/17EAqhVca4CYMUjyTtJVcm5bA5Kv7n3gTVjqRBYE546E/viewform?c=0&w=1>)
5. After the volunteer stakeholder had the opportunity to review the prototype, the following questions were asked:
   1. Could they use it without help?
   2. What were the problems?
   3. Did they like it?
   4. What did they like most/ least?
6. Compiled the list of improvements and enhancements and added into Sprints 2-5 for Initial Release 1.0
7. Provided a demonstration to the Product Owner and stakeholders at Sprint Review for 100% acceptance

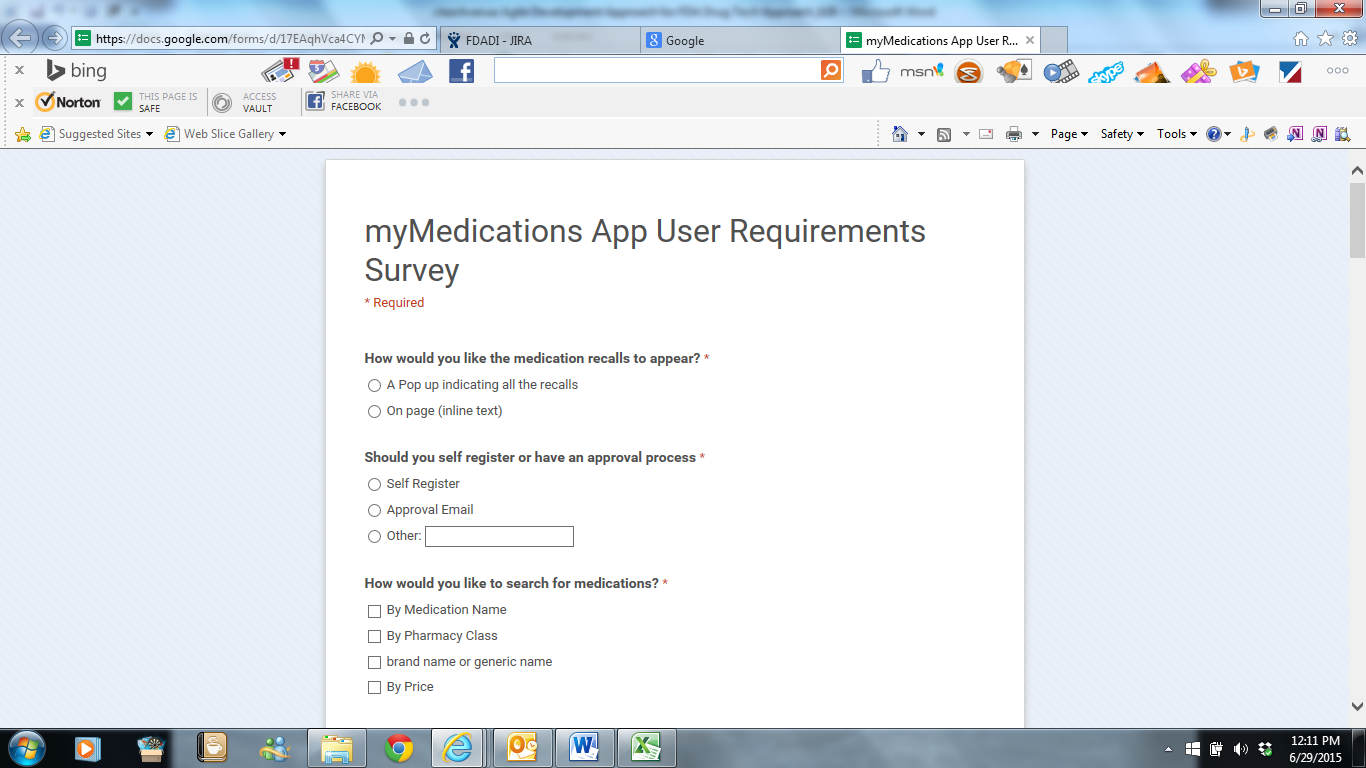


Exhibit 2.0-2 Usability Requirements gathering through User Surveys.

Lastly, we requested volunteers to review the developed application for User Acceptance Testing (UAT), the following questions were asked after completion of the UAT:

1. Did it meet documented and accepted requirements?
2. Suggested enhancements and improvements

We compiled the list of bugs and enhancements from the Product Backlog into future releases (See Appendix 1 for Full Jira Export). We received approval for Product Owner and UAT participants for Release 1.0.

2.1 Development Process

Some key aspects of our iterative agile development processes that we used for developing and deploying the myMedications App are demonstrated in the charts below from JIRA.

* Sprint 1- We developed the initial project plans and compiled the user stories, tasks and chores into JIRA.
* Sprint 2- We accepted too many requirements against the Sprint and were unable to complete all of the stories/ tasks assigned for the Sprint.
* Sprint 3-5- We were cautious during our Sprint Planning sessions and worked to document the story points (LOE) and accepted minimal requirements per Sprint to ensure that we would be able to match the Release Requirements. We had many defects that were resolved during these sprints.
* Sprint 6- We accepted only bugs and subtasks that did not have LOE/ story points associated.
* Sprint 7- We finalized the release by working with our focus groups and developing the improvements and enhancements requested by the end user.

As seen in Exhibit 1.0-2, after the customer stories are placed in JIRA, the development process starts with a developer selecting a Task assigned to them in JIRA and moving it into the 'In Development' state.

Code and tests are created to implement the selected tasks and then committed to GitHub with JIRA tags in the commit message to link the GitHub commit to the JIRA ticket. Bamboo, the continuous build/integration server, will detect the commit and trigger a build which includes JUnit tests, Selenium tests, and Clover code coverage creation. If successful, Bamboo will start the deploy process and deploy to a Docker container located in the AWS Cloud. More information on the Bamboo build and deploy steps are shown later in Section 4.0.

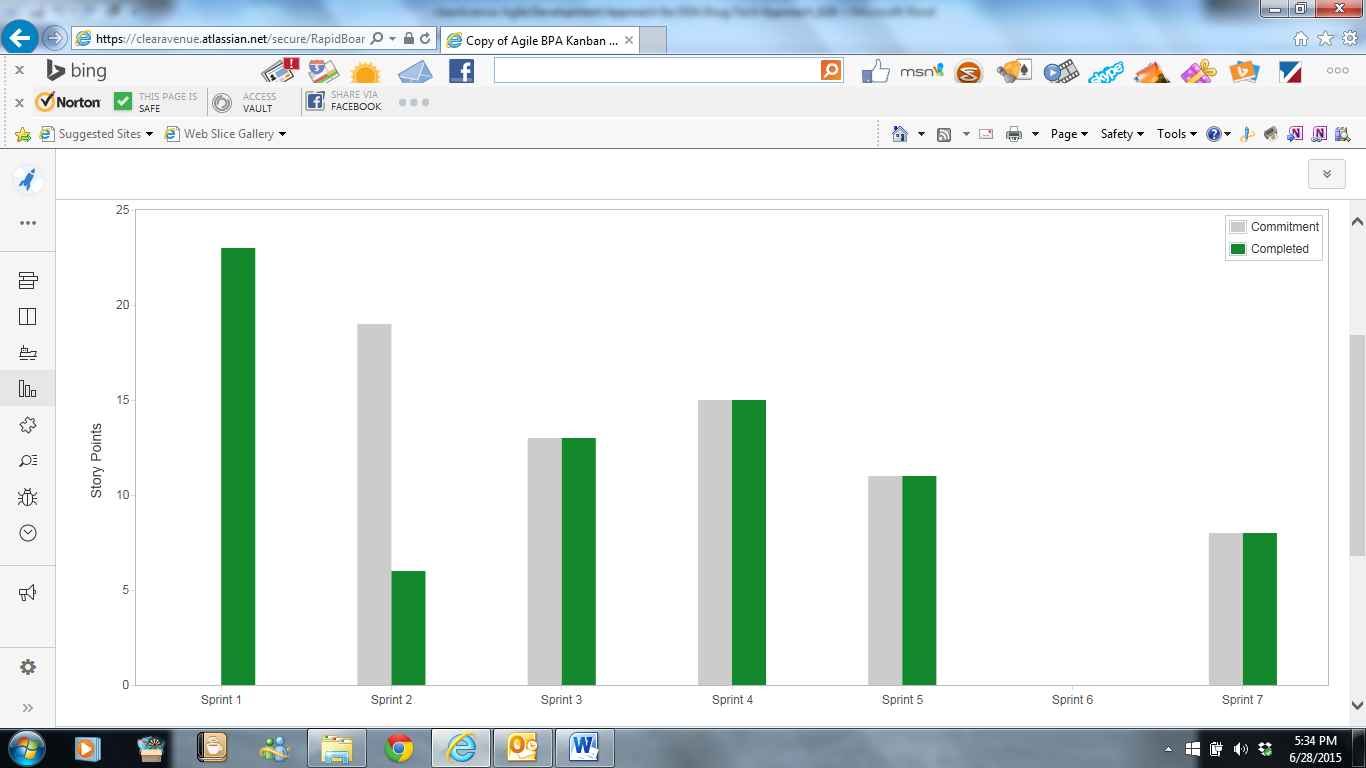


Exhibit 2.1-1: We used 7 sprints to develop incremental functionality and to deploy it to the AWS cloud. The development sprints also included continuous integration, automated testing and continuous deployment

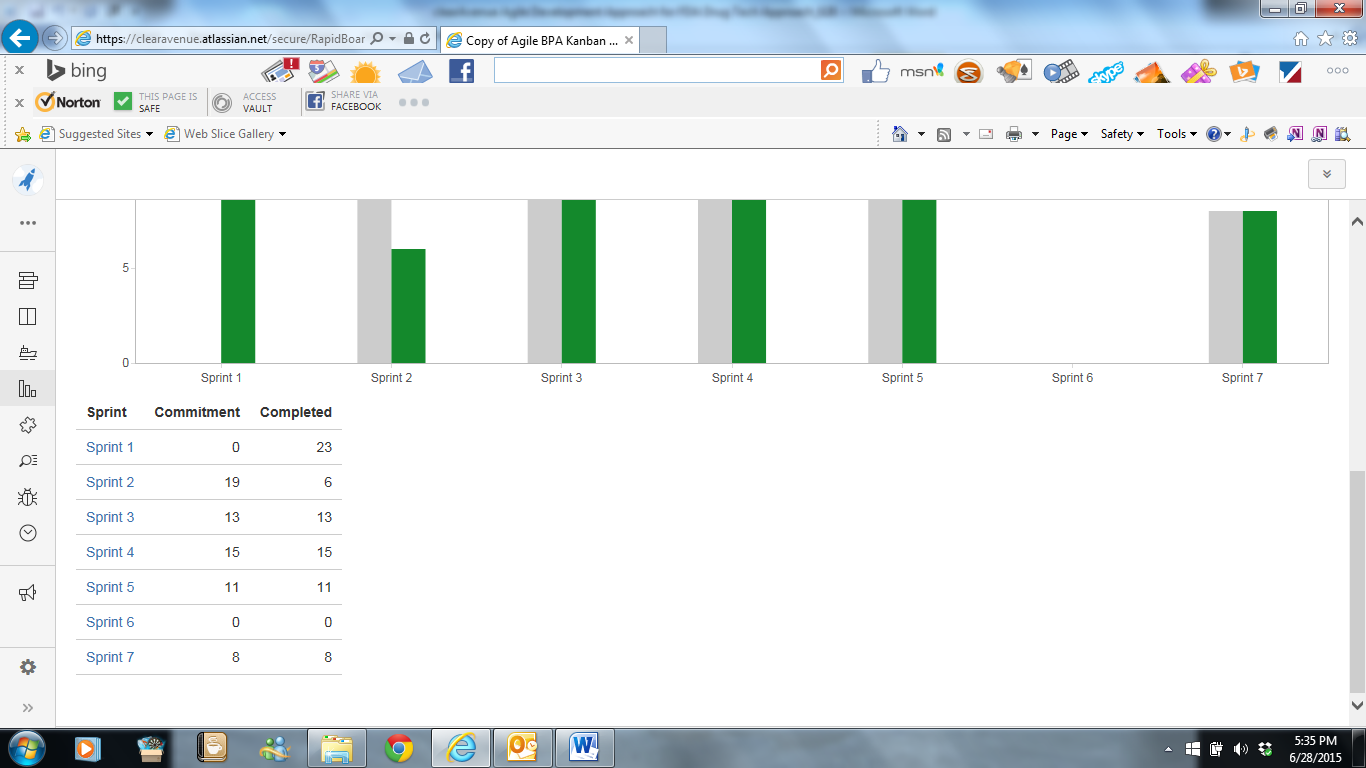


Exhibit 2.1-2 Additional details of the various sprint commitments and the completion of assigned stories, tasks and chores

2.2 Monitoring Progress

At any point in time, the total work remaining to reach a goal can only be estimated. The Product Owner tracks this total work remaining for every Sprint Review. The Product Owner compares this amount with work remaining at previous Sprint Reviews to assess progress toward completing projected work by the desired time for the goal. This information is made transparent to all stakeholders.

Various trend ***burn down, burnup*** and other projective practices have been used to forecast progress. Exhibit 2.2-1 shows the Epic burn down for the myMedications application over 7 sprints. Exhibit 2.2-2 shows the cumulative progress in delivering the application over 7 sprints.

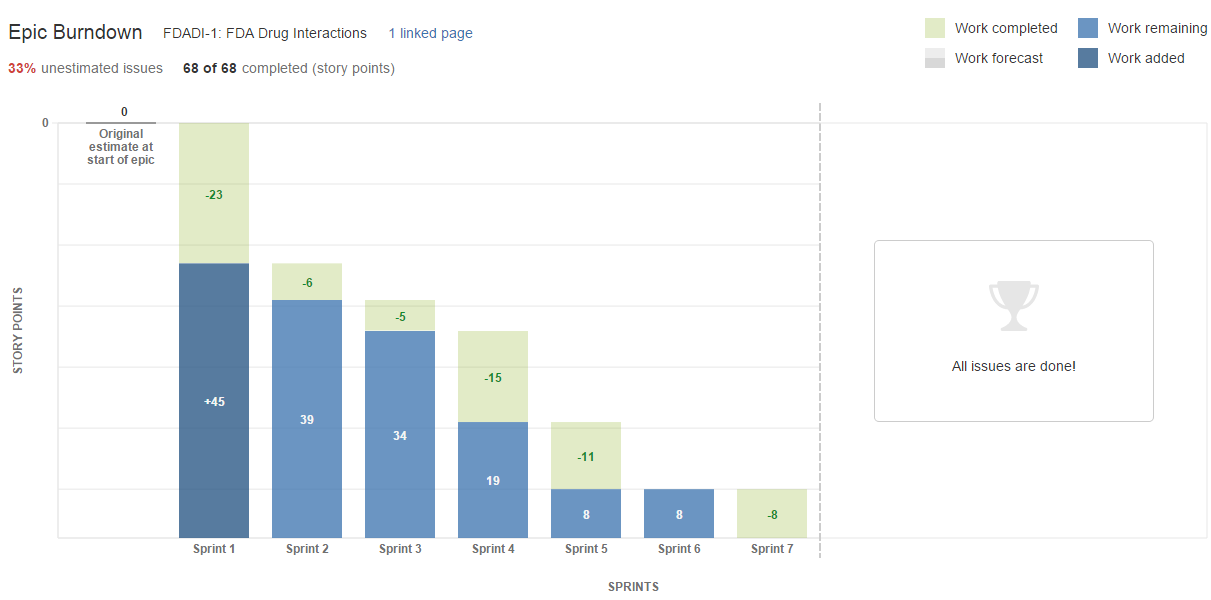


Exhibit 2.2-1- Tracking progress on completion of the various user stories and obtaining feedback

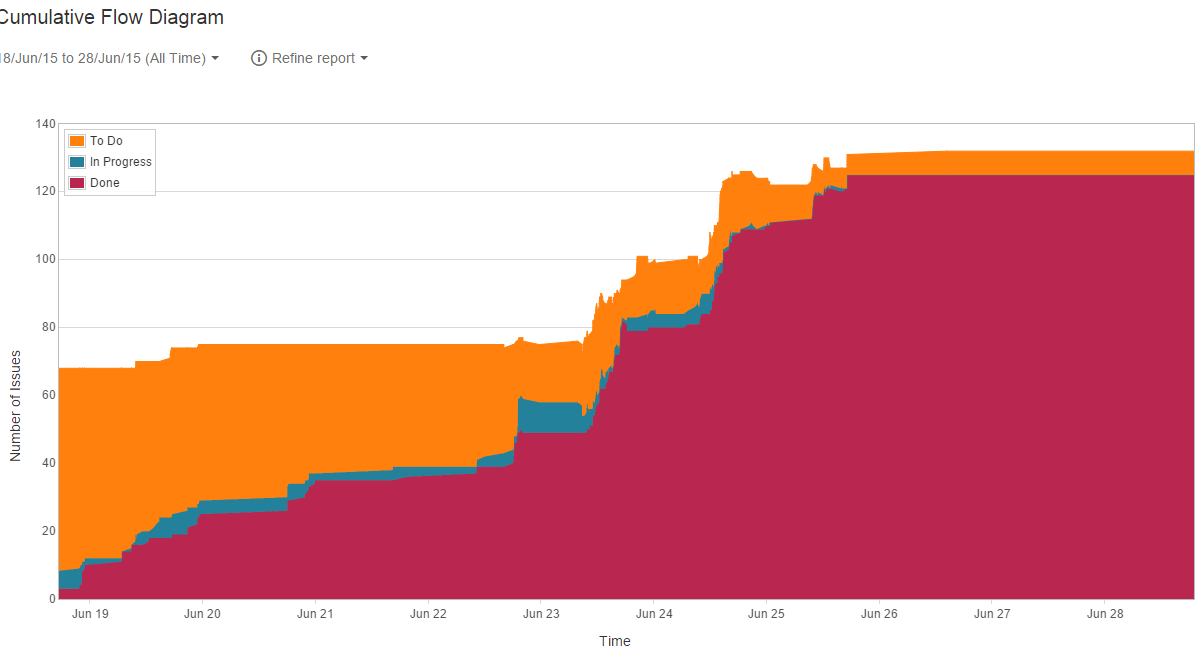


Exhibit 2.2-2: Cumulative progress over the allocated time in various sprints

# 3.0 Tools Used

|  |  |
| --- | --- |
| Tool Used | Functionality |
| JIRA | Planning Sprints, Tracking Progress, Release Management, Defect/Issue Tracking |
| Bamboo | Continuous Integration Server, Build Management, DevOps |
| Confluence | Collaboration, Team Communication, Team Wiki |
| Maven | Continuous Integration, DevOps |
| Nessus Scanner, Nagios | Continuous Monitoring |
| Selenium | Web Application Testing |
| Junit, FindBugs, Clover | Unit testing, Testing Code Coverage |
| Docker | Container |
| Spring MVC, Bootstrap, Logging SL4j, Apache Tomcat | Middleware Infrastructure for myMedications |
| AWS | Platform as a Service (PaaS for Deployment) |

Exhibit 3.0-1 shows the tools that are used for the DevOps environment for myMedications application.

# 4.0 Continuous Integration, Deployment and DevOps

The myMedications application utilizes JIRA to track all user stories, tasks, bugs and team progress and Bamboo with GitHub to provide continuous integration and delivery to tie automated builds, tests, and releases together in a single DevOps workflow. JIRA, Bamboo, and GitHub work well together to provide a fully traceable development pipeline.

Bamboo performs continuous integration in 2 phases: Build and Deployment.

Bamboo will poll the GitHub repository and will trigger a build when a GitHub commit is discovered. Bamboo will check out the source code, perform a Maven build, run JUnit tests, and a Clover code coverage report. If the build fails, the developers are notified by email of the failure, the responsible party, and the corresponding Git commit that triggered the failed build. If the build passes, Bamboo will trigger the deployment process.

The Bamboo deployment process will check out the source code, perform a build, run JUnit tests, and perform a static analysis using FindBugs. If the deployment build fails, the deployment is stopped and the developers are notified by email. If the deployment build passes, Bamboo will deploy the myMedications WAR file onto a running Docker container-containing running instances of Apache Tomcat and MongoDB-located in the AWS Cloud.

Exhibit 4.0-1 shows the continuous Integration and deployment (DevOps) for the epic development and deployment.

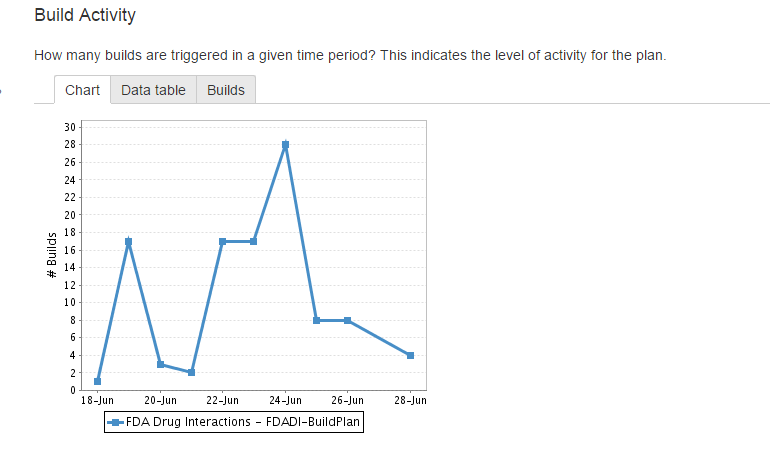
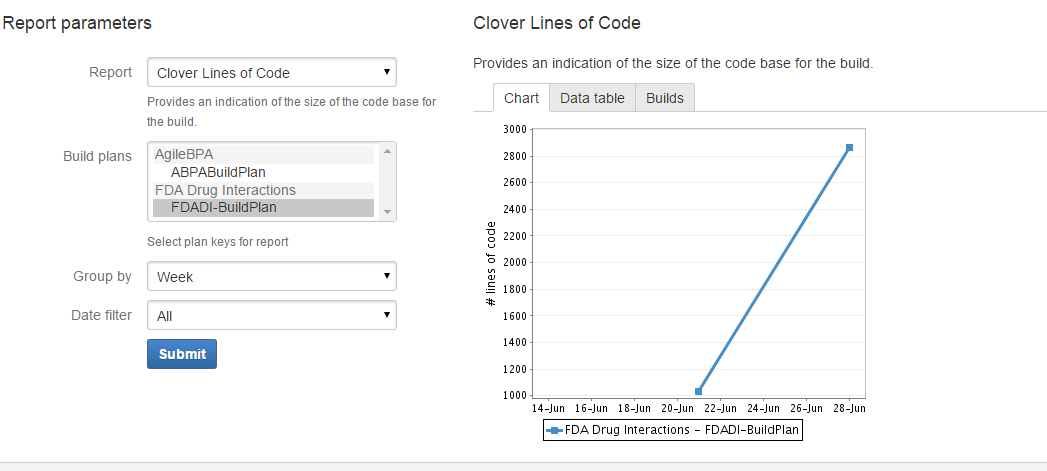


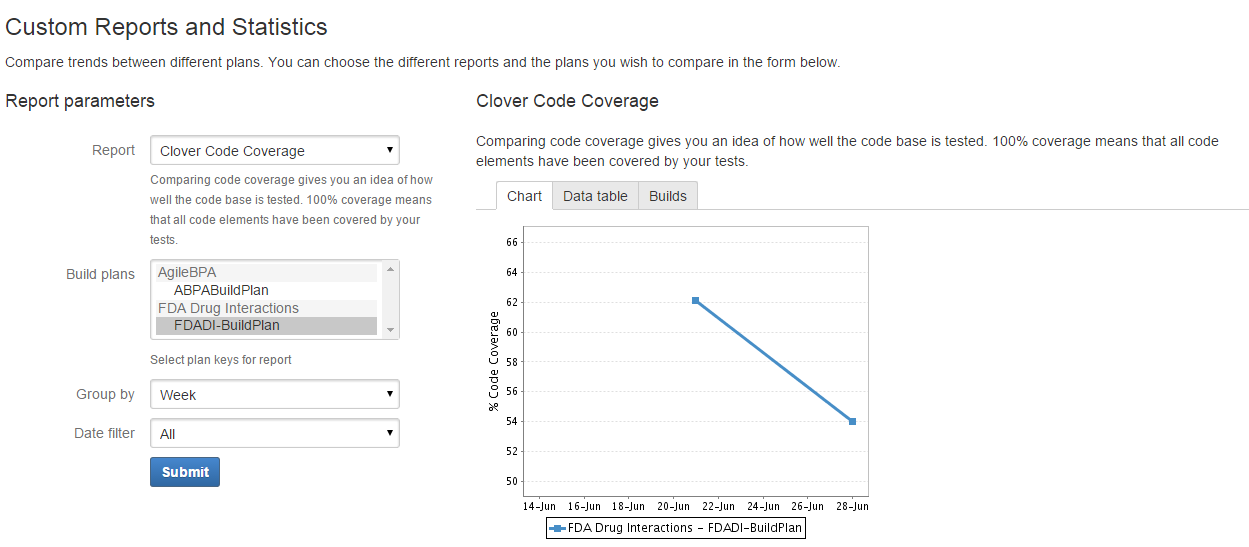
Exhibit 4.0-1 shows the continuous integration and deployment (DevOps) for the Epic development from 18 Jun to Jun 28. The graph shows the daily builds and automatic deployments to the AWS infrastructure.

# 5.0 Test Automation

We have used the Bamboo Continuous Integration server together with Apache Maven and a variety of test tools such as JUnit, Selenium, FindBugs and Clover for test automation. Maven uses a pom.xml file in the GitHub repository to contain information about the project and configuration details used to build the project. This configuration specifies all dependencies and plugins used to create the myMedications WAR file. The myMedications projects pom.xml includes plugins to perform JUnit tests and dependencies to run Selenium integration tests. The Maven build automatically runs the myMedications unit tests using JUnit and web integration tests using Selenium. The actual tests are part of the source code in the GitHub repository. For additional details please see the document titled myMedications Test Automation Results.pdf in the Git repository under documents.



**Exhibit 5.0-1 shows the increase of the myMedications code base over time. The tests are run automatically as part of the Maven Build and deployed to the IaaS infrastructure on successful build.**



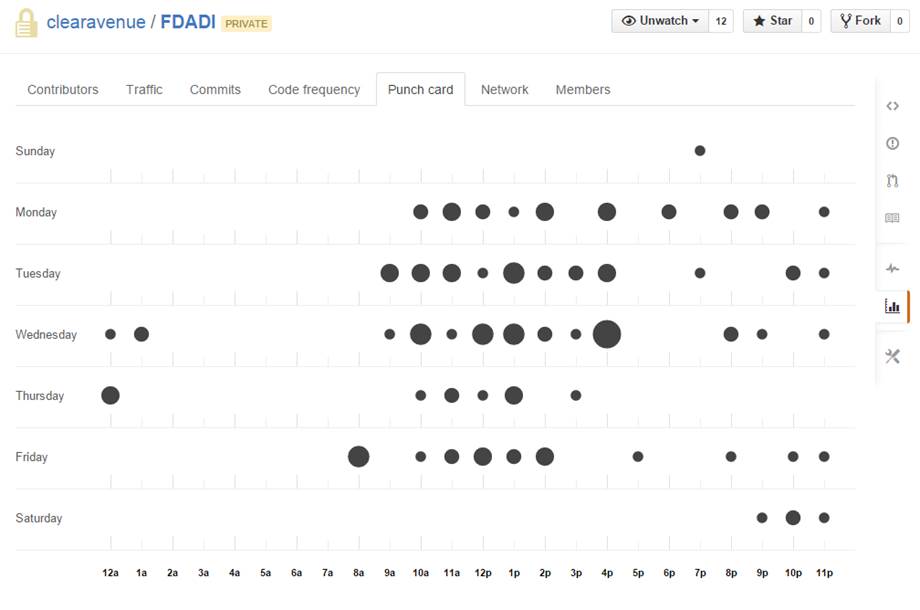
**Exhibit 5.0-2 shows myMedications test coverage over time. The tests are run automatically as part of the Maven Build and deployed to the IaaS infrastructure on successful build.**

# 6.0 Continuous Deployment

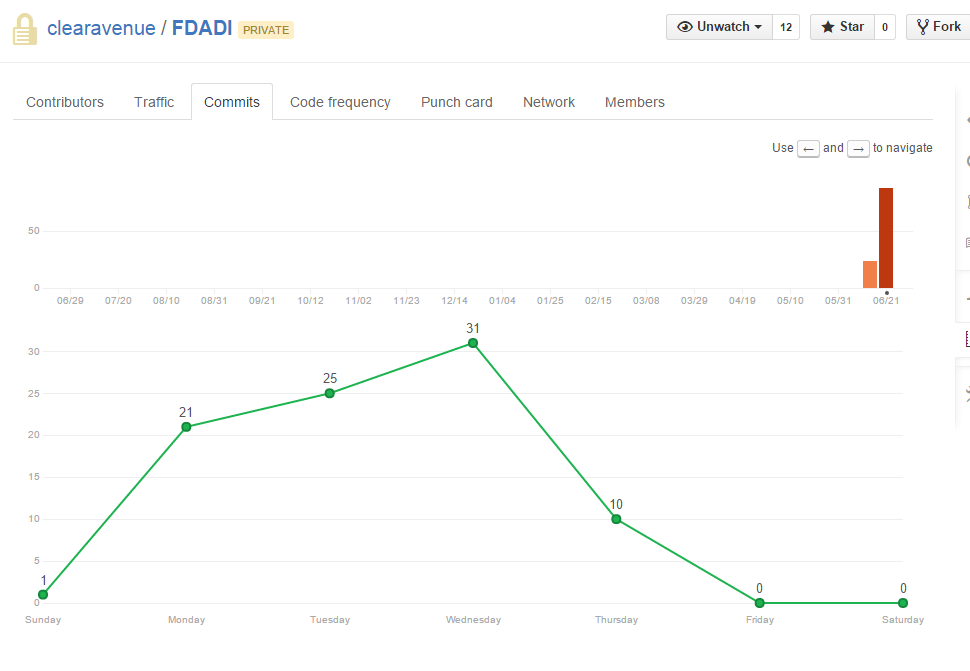
On successful completion of test cases, our continuous integration server performs continuous deployment to the AWS cloud. The following GIT reports show the Continuous Deployment approach used by clearAvenue to build the myMedications App.



Exhibit 6.0-1 Continuous Integration and Deployment Snapshot as viewed from the Git repository for the development and deployment of the myMedications Application.



**Exhibit 6.0-2 Continuous Integration and Deployment details for the myMedications App. This Exhibit shows the details of the commits performed by the DevOps team.**



**Exhibit 6.0-3 Continuous Integration and Deployment details for the myMedications App. This Exhibit shows the details of the commits performed by the DevOps team.**

# 7.0 Continuous Monitoring

clearAvenue has utilized a combination of AWS infrastructure monitoring in conjunction with Nagios Application monitoring to cover the availability aspects of myMedications. We also intend to run periodic Nessus scans to identify the vulnerabilities and remediate the vulnerabilities immediately. We have ensured that all the components used in myMedications are patched and no known vulnerabilities exist. We perform scheduled yum updates every two weeks to ensure that the various system components are at the right patch levels. We also have utilized OWASP and other best practices in the design and implementation of the application.

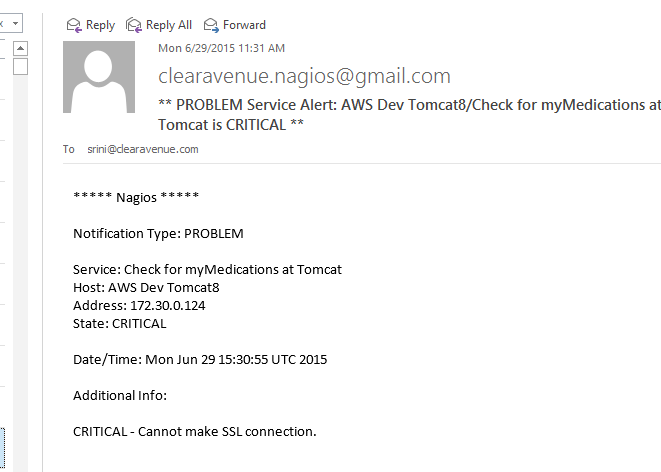


Exhibit 7.0-1: As part of the continuous monitoring (availability monitoring) we have deployed Nagios to monitor the myMedications Application. This monitoring is in addition to infrastructure monitoring provided by AWS.

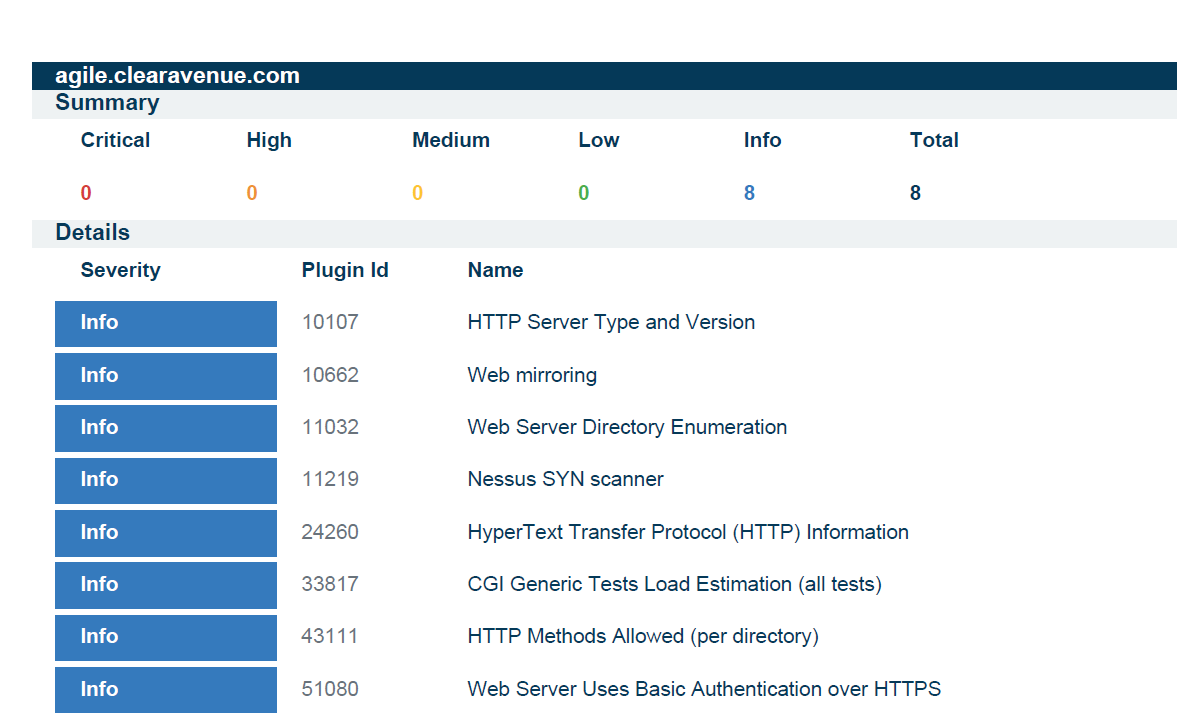


Exhibit 7.0-2: As part of the continuous monitoring we plan to run periodic (weekly) vulnerability scans of the myMedications Application. Exhibit above shows that the application has no critical, high, medium or low vulnerabilities.

Additional details can be found in the myMedications Vulnerability Scan.pdf under the documents directory in the Git repository.

# 8.0 User Acceptance Testing (UAT)

We performed User Acceptance Testing (UAT) to verify the business functionality and proper functioning of the system. For the myMedications Application, the product owner, our target SMEs as well as our alpha/beta customers performed the UAT testing. The UAT was performed at the end of every sprint and user stories were re-opened, defects identified and issues were generated to ensure product quality.

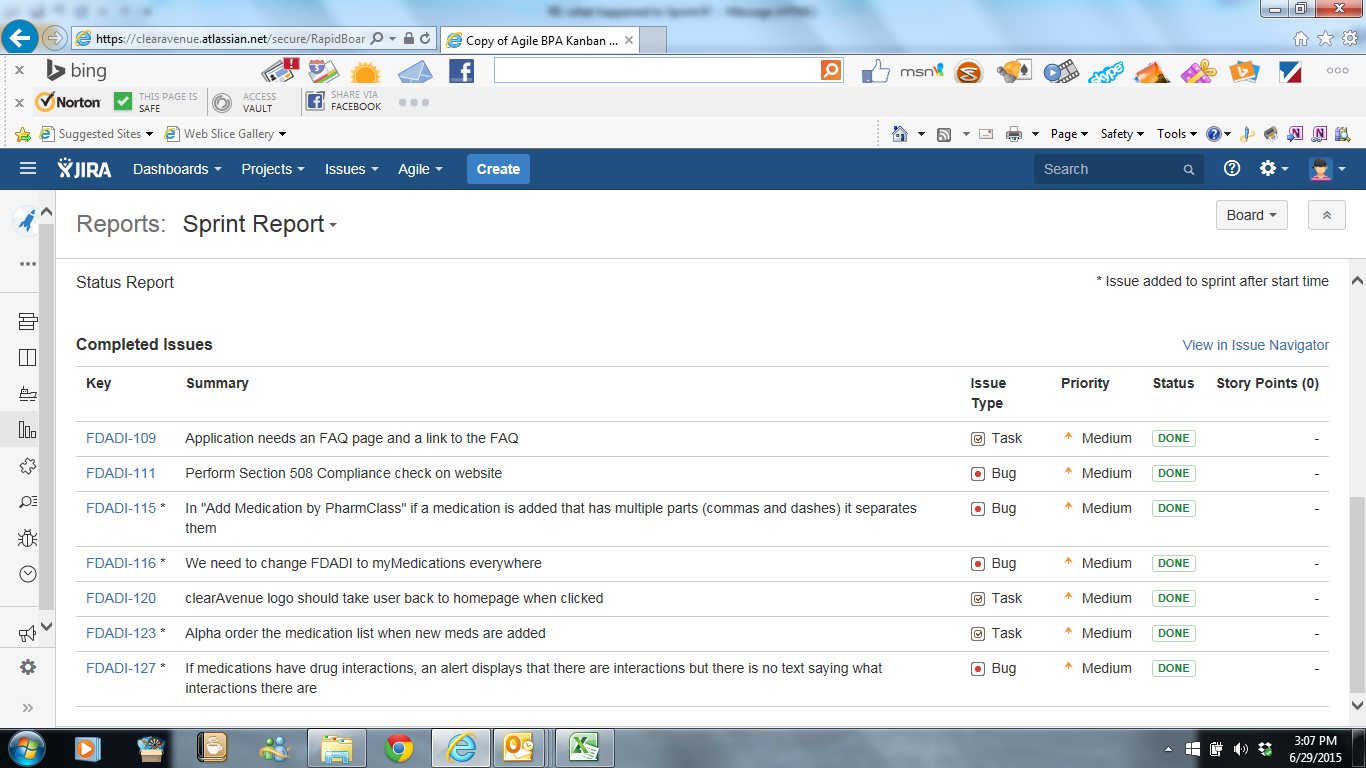


Exhibit 8.0-1: Defects and Issues generated during the User Acceptance Testing.

8.1 Testing on Mobile Devices

The Application was tested comprehensively on mobile phones and tablets. We have tested myMedications on a variety of browsers such as Chrome on Android Phones and Tablets, and Safari on iPads. Screen shots of these devices are included below. Usability testing was also performed to check the aspects of responsive design.

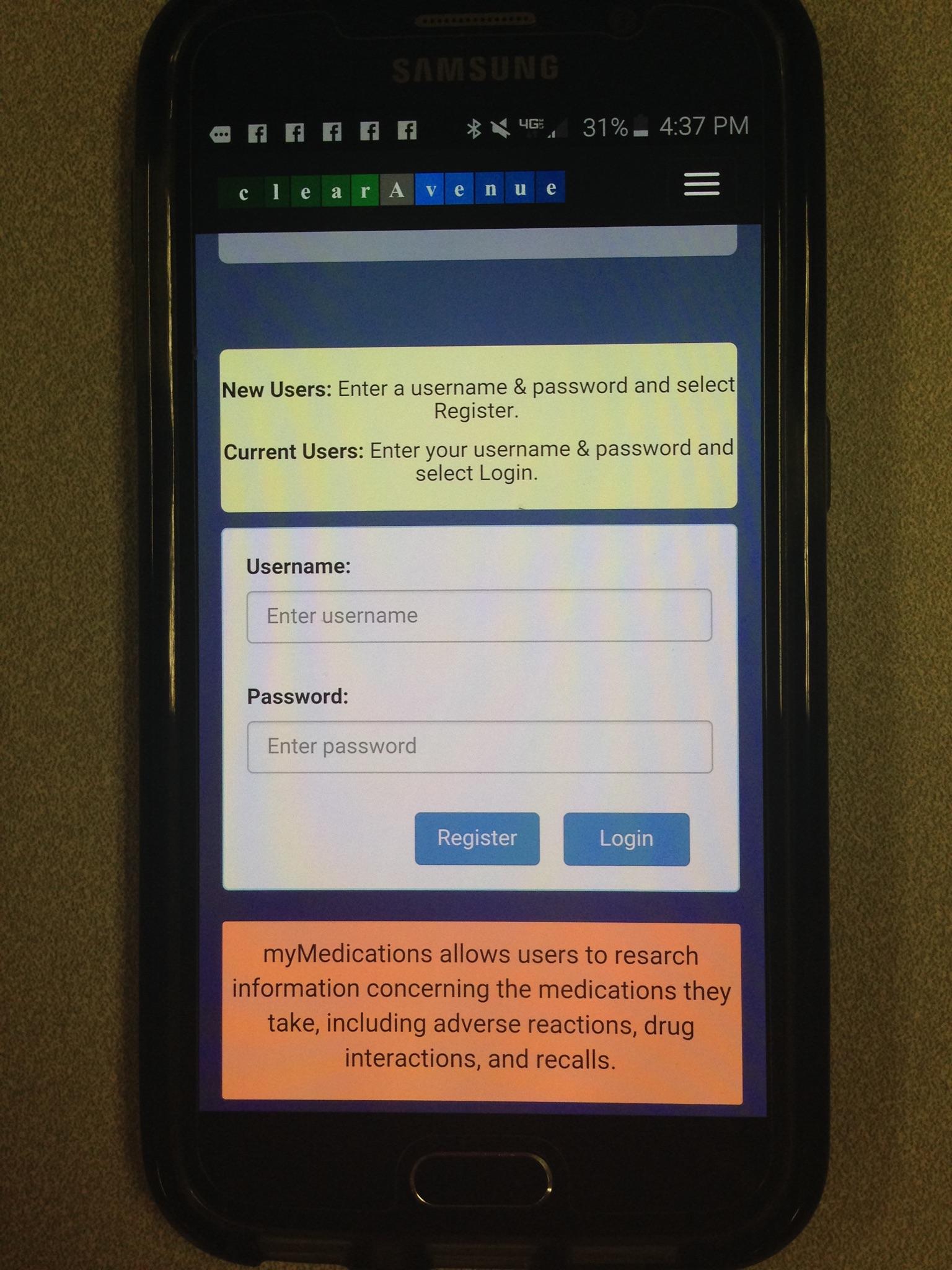


Exhibit 8.1-1 myMedications App Testing on Android Phone-Screen 1

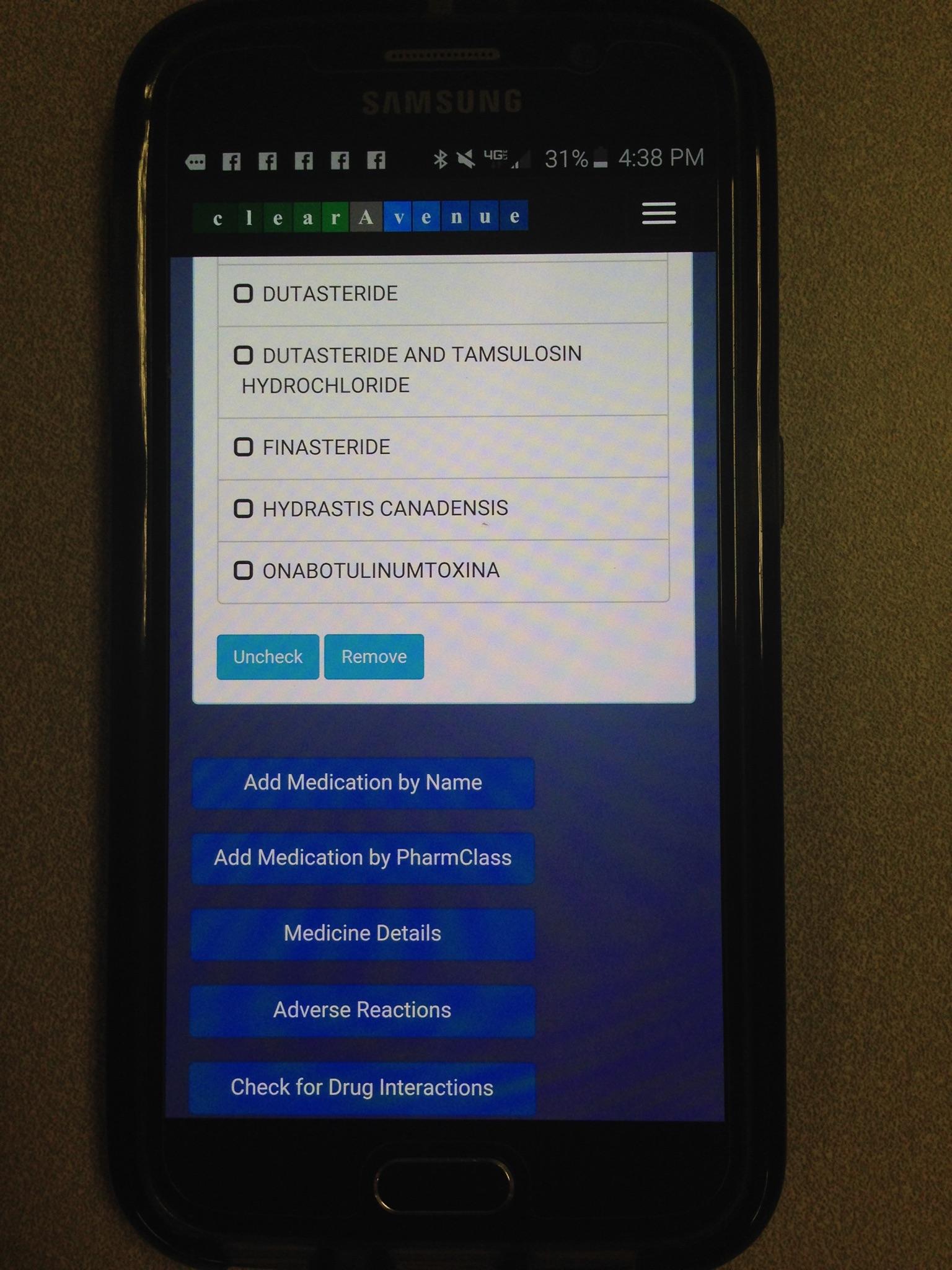


Exhibit 8.1-2 myMedications App Testing on Android Phone-Screen 2

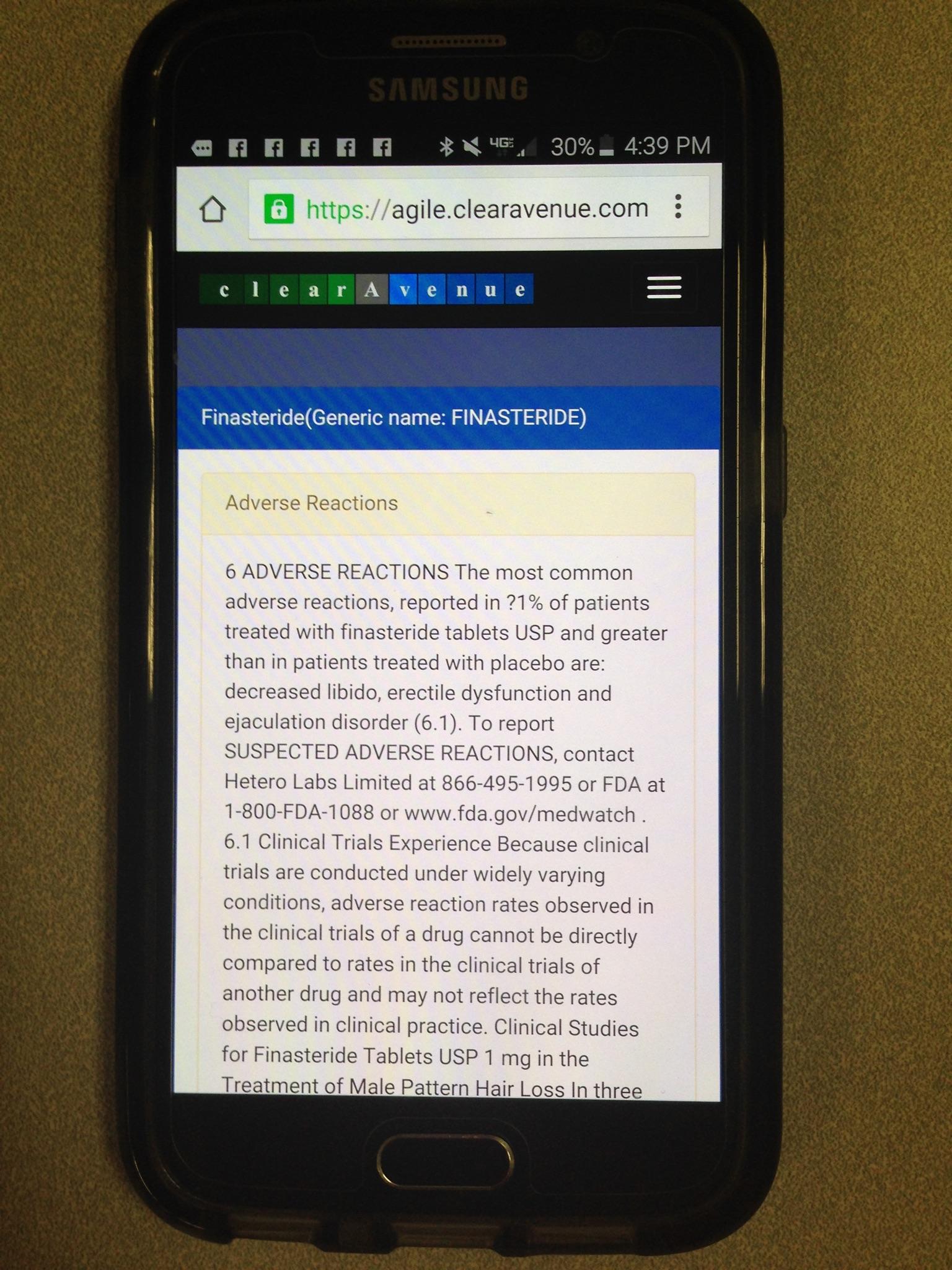


Exhibit 8.1-3 myMedications App Testing on Android Phone-Screen 3

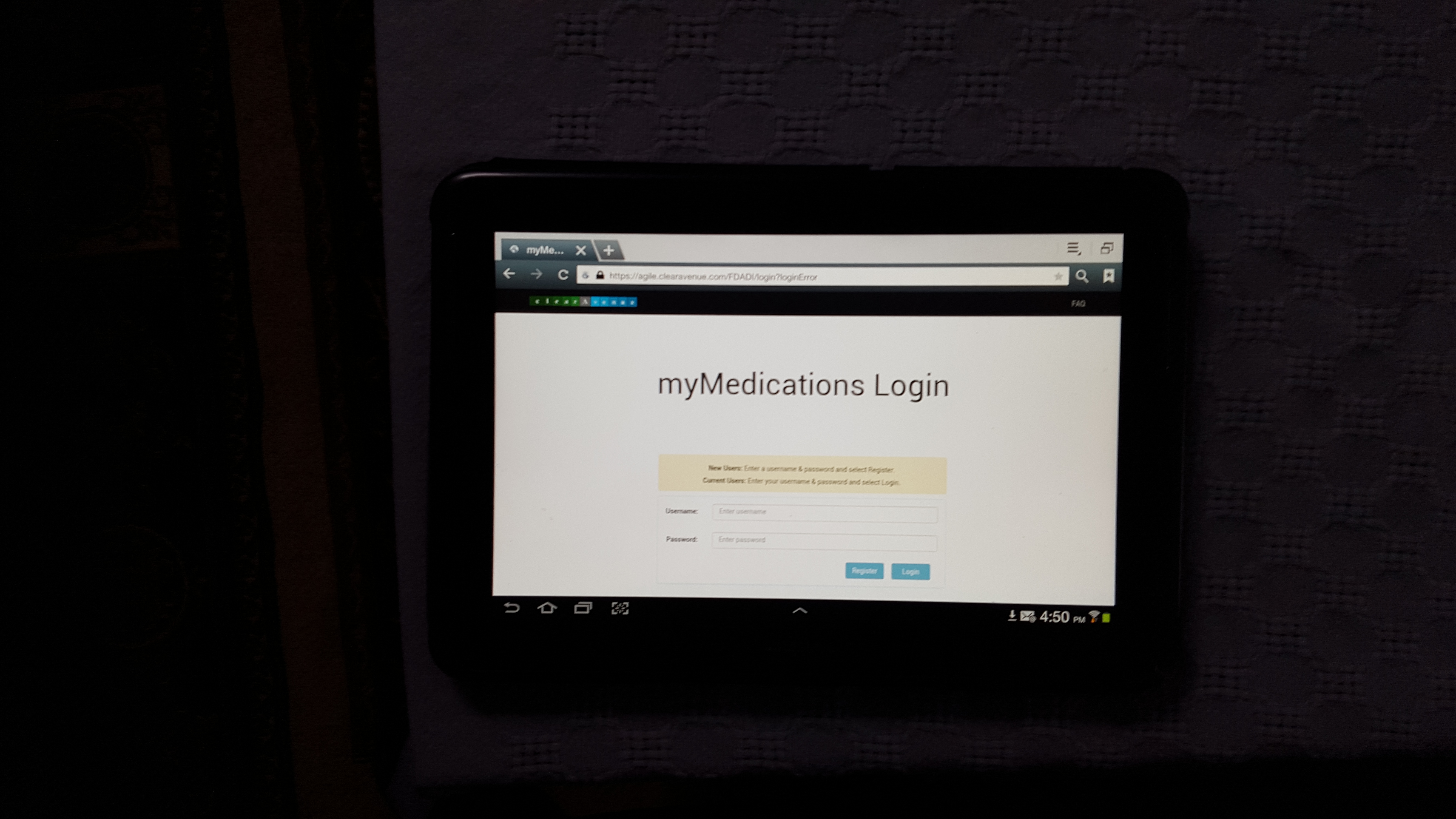


Exhibit 8.1-4 myMedications App Testing on Android Tablet -Screen 1

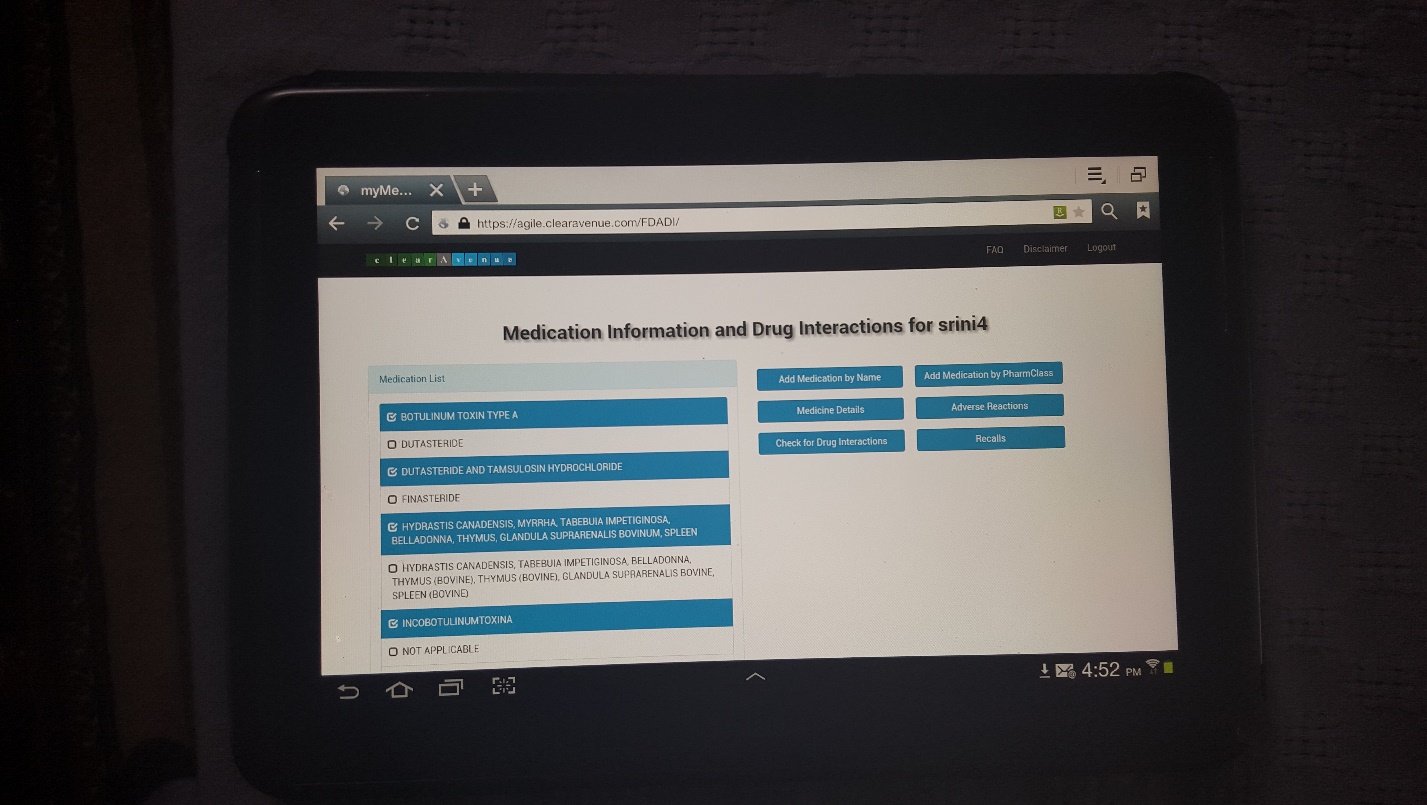


Exhibit 8.1-5 myMedications App Testing on Android Tablet -Screen 2

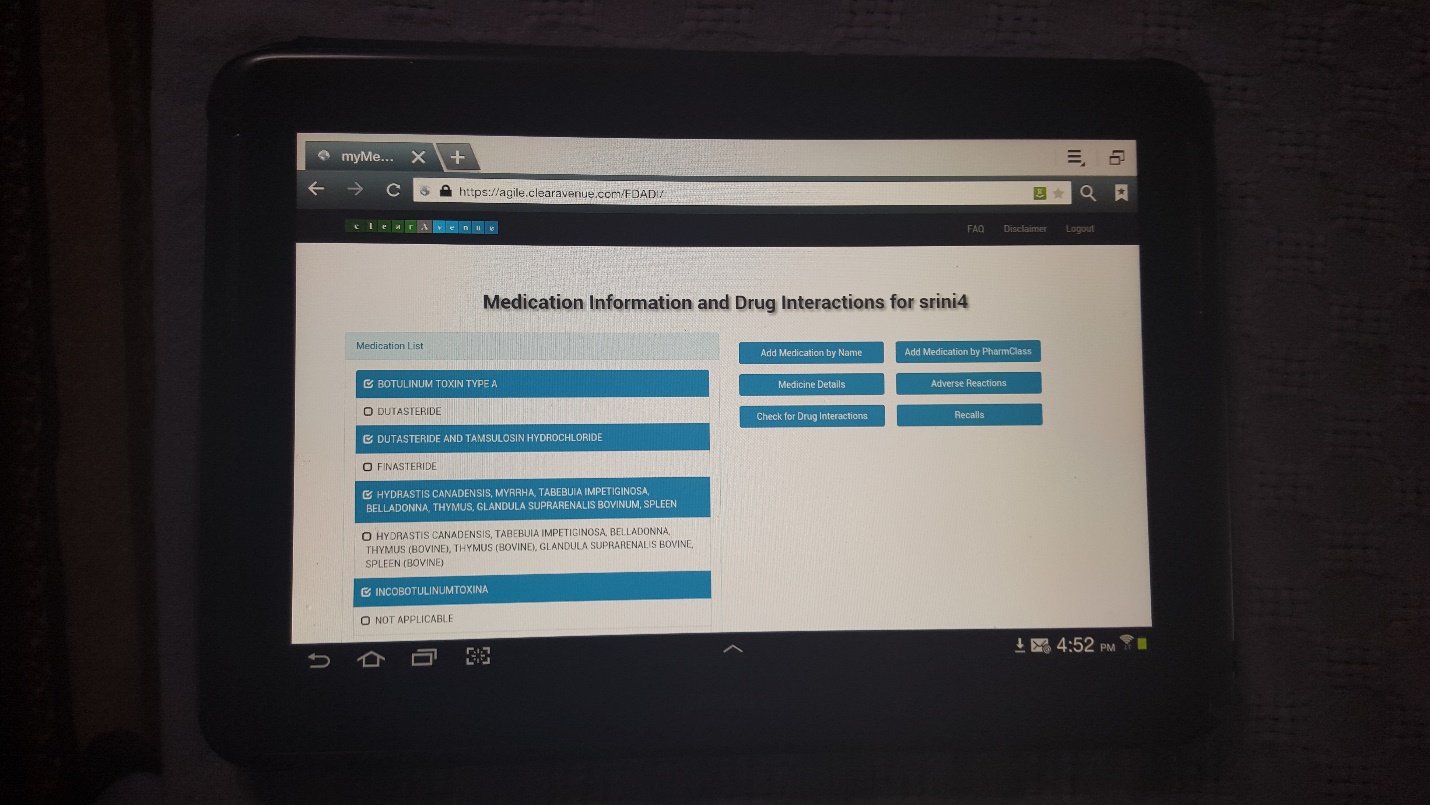


Exhibit 8.1-5 myMedications App Testing on Android Tablet -Screen 3

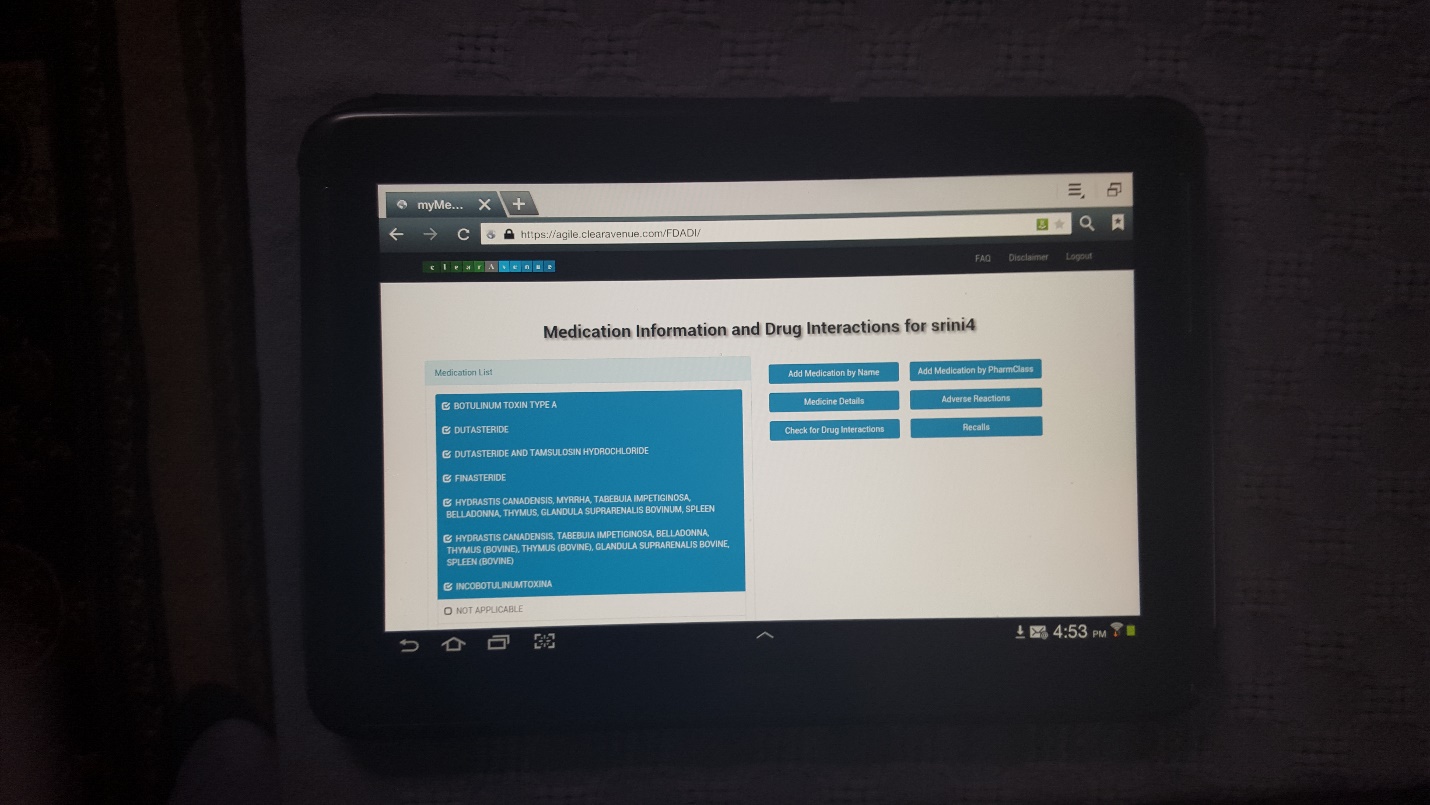


Exhibit 8.1-6 myMedications App Testing on Android Tablet -Screen 4

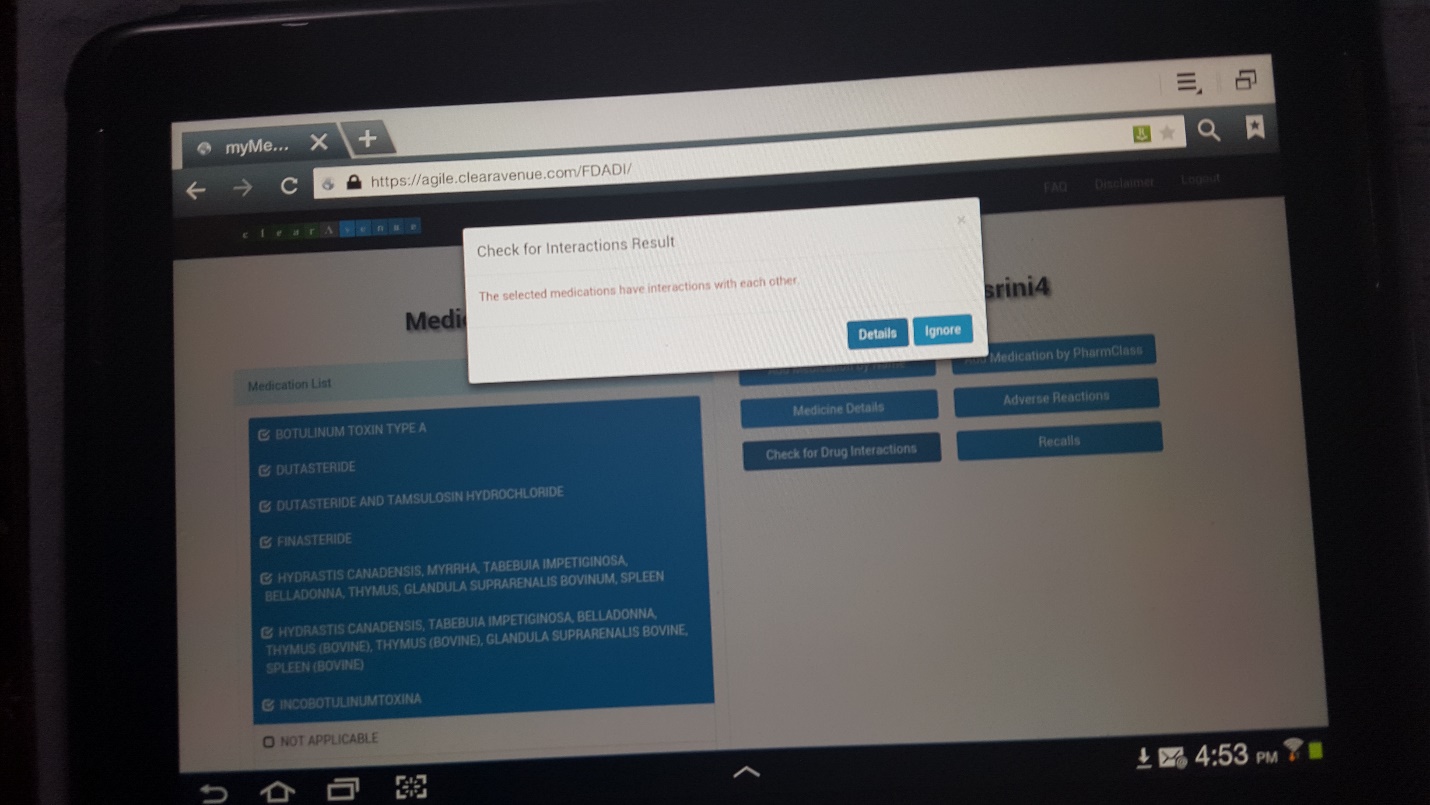


Exhibit 8.1-7 myMedications App Testing on Android Tablet -Screen 5

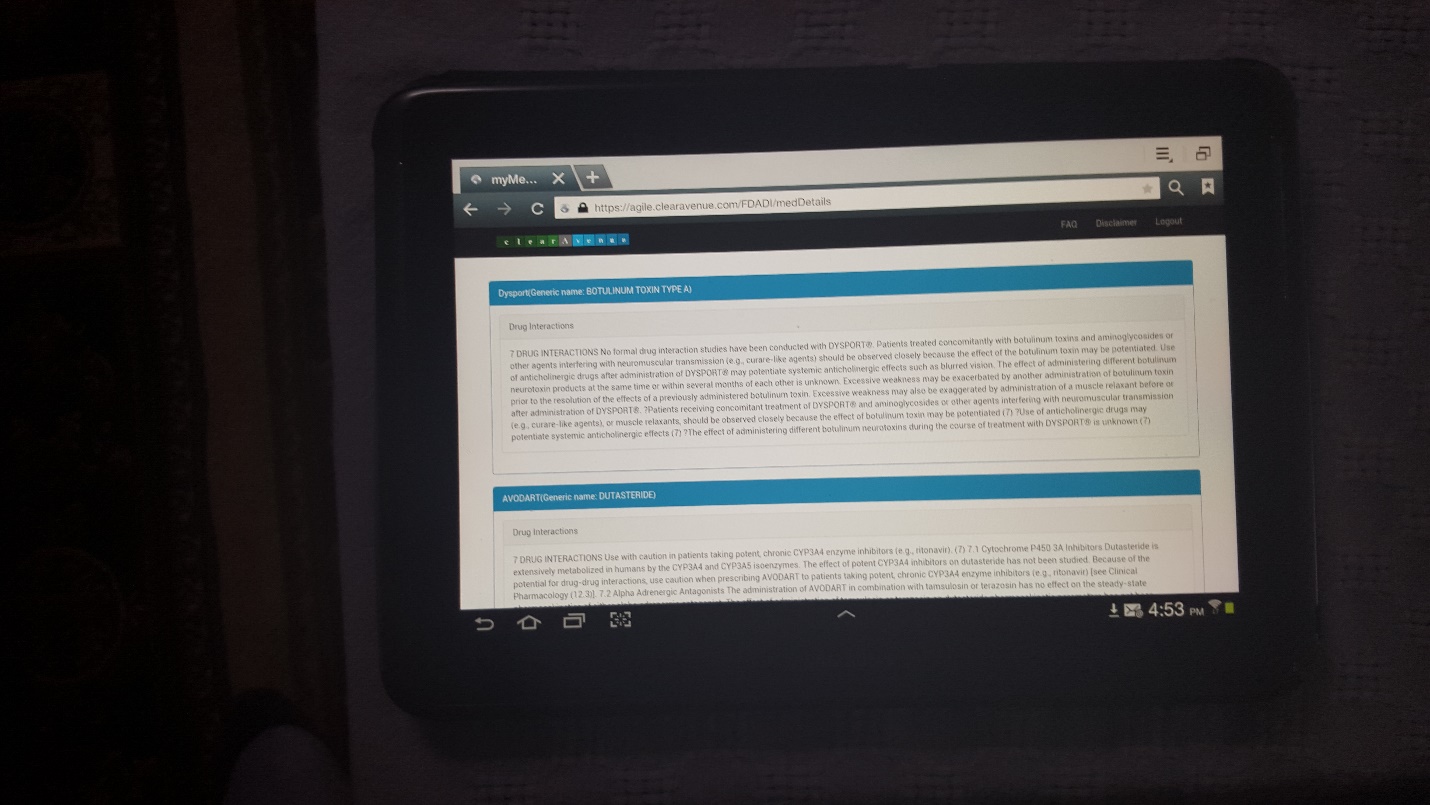


Exhibit 8.1-8 myMedications App Testing on Android Tablet -Screen 6

# Appendix 1- JIRA Export of the User Stories, Tasks and Chores

See myMedications (FDADI) user stories.xls in the documentation directory.