ABS – Post Mortem Report

May 29, 2017

Daniel Caddaye - s3383022, Cory Patterson - s3600197, Sohum Raut - s3377766, Marco Riotto - s3606584

Contents

Description

Name: Appointment Booking System
Client: RMIT
Project Manager: Daniel Caddeye
Solutions Architect: Cory Patterson
Start Date: Feb 28, 2017
Completion Date: 28th May 2017

Project Overview:

The Charter:

We were tasked with creating an application which allowed businesses to manage their appointment bookings with their customers. The software had to be a robust and well tested program which could handle multiple types of users such as business owners and customers, as well as be able to store many different details of the business and their bookings.

Success Criterion:

- Login as a customer
- Register as a customer
- Login as a business owner
- register as a business owner
- Save and load business data
- View, Book and cancel Bookings for a business as a customer
- Business owner can add employees
- Business owner can add available booking times
- The System MUST be a GUI (Graphical User Interface) and not just command line application
- The system must be scalable and deployable

.

Performance

Key Accomplishments:

Most Team members worked well together and got the tasks at hand done in the given timeframe which was good.

The code in which the project was written was done in a very high standard we believe and are proud of the framework we have created, it is easily extendable and open source.

Particularly useful was using the Trello website to keep our sprints organised and it was simple to see what tasks were left to do. Also useful was Slack the messaging app, it felt a lot more professional to be using a software designed for team messaging rather than using something too personal such as Facebook messenger, and we also believe that emails are becoming dated to these modern software development teams.

Highlights of the project was getting the GUI up and running, and having the project as an executable java file once built gave us all a sense of accomplishment.

Key Problem Areas:

We had some trouble at the beginning of the project with managing our git repository, but once we found out there was something called 'branch protection' which meant a pull request must be reviewed by another team member, the problem was solved straight away as issues were caught and addressed before merging into the master branch of the project.

We found the sprints to be challenging at first but after the first one we could work much better once we fully understood what was going on.

The effects of the problems did delay us slightly in our schedule but this was easily caught in the following sprint and not much changed overall.

Technical Challenges were mainly getting eclipse to integrate with git on some of our older laptops and computers. Seemed to come up time and time again with a slightly different way it was broken and a completely different fix. But in the end, we prevailed.

Risk Management:

Risks that were mitigated was merge risks that may reintroduce bugs that we previously fixed, we limited this by having a process in place where we would first update our working branch with the changes from the master and then make a pull request to the master once we were sure that the functionality still passed our JUnit tests.

Risks that still need to be managed would probably be the portability of the software, we did do some testing on both Mac OS and Windows but did not test on Linux Based operating systems although we are confident that JAVA runs on most operating systems without a hitch.

Overall Project Assessment:

We would rank our project and team overall as probably an 8.5 out of 10. We thought that we worked well together to achieve a reasonable amount of work as per the project outline and had fun doing it.

Additional Comments:

Our Product Owner (Tutor) Mohammed was very good to work with and provided us with the right direction for a successful project. And the processes used were highly insightful into what a real development team would work with.

Key Lessons Learned

Our Key Lessons Learned would be:

- **Use Test Driven Development where possible**

This turned out to be a great process with such a speedy development lifecycle

- **Always Use Branch Protection on Git projects**

Although it was sometimes a pain, having another set of eyes to take a look at your code not only means that you will push harder to bring up your standard of coding, it also means little things that you might have missed have a chance to be picked up before they become a real problem.

- **Comments are Great!**

Having nicely formatted code and object orientated programming although is simple when writing it, looking back and looking at other people's code was sometimes a chore to figure out what is going on, especially if you have to debug it. Therefore, we found comments were immensely useful as everyone knew what was going on without any trouble at all.

Future Considerations:

In the Future, the project could be extended to do many things and even turn into a full Business management software if enough time and money were to be put in it. With the current code structure, it is quite easily extendable to do what we need it to do and could allow for extra 'plugins' in the future which could achieve new functionality such as this. The main thing to look out for in future is maintaining the current quality of code and using the same standard of processes to ensure a great product for the user. The project could take many forms in the future and it should be worked on closely with the product owner.

Maintenance of the code should be done often to try and optimise the size and speed of the program although as it is currently, the app is very fast and efficient. although when the data being stored becomes quite large there may be some considerations on changing our storage protocols.

Yet to be completed is the functionality of adding the employee hours, currently a business owner must make a separate appointment time for each available time an employee can be booked and in the future, we want to add a feature where a business owner can simply add a length of time and the system turns this into available times automatically.

Nothing else of our product is yet to be completed.

Self-Assessment

Performance against project goals/objectives	1	2	3	4	5	6	7	**8**	9	10
Performance against planned schedule	1	2	3	4	5	6	7	8	9	**10**
Performance against quality goals	1	2	3	4	5	6	7	**8**	9	10
Performance against planned budget	1	2	3	4	5	6	7	8	9	**10**
Adherence to scope	1	2	3	4	5	6	**7**	8	9	10
Project planning	1	2	3	4	5	6	7	8	9	**10**
Resource management	1	2	3	4	5	6	7	8	**9**	10
Project management	1	2	3	4	5	6	7	8	**9**	10
Development	1	2	3	4	5	6	7	8	**9**	10
Communication	1	2	3	4	5	6	7	**8**	9	10
Team cooperation	1	2	3	4	5	6	7	**8**	9	10
Project deliverable(s)	1	2	3	4	5	6	7	8	**9**	10