Software Engineering Group Project
Final Report

Author:	Jasper Crabb [jac127], Archie Malvern [arm36]
Config Ref:	FRGroup18
Date:	10th May 2023
Version:	1.1
Status:	Release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

CONTENTS

1. INTRODUCTION

1.1 Purpose of this Document

The purpose of this document is to discuss the whole project, the processes that we went through while working, the final state of our project and the performance of each team member.

1.2 Scope

This document is a report regarding the project as a whole so it is recommended that you have read all of the documents regarding the project and are familiar with the final state of the project before reading this.

1.3 Objectives

The objective of this document is to give the client a comprehensive account of how our team worked to create the project and what the final state of the project is.

2. MANAGEMENT SUMMARY

2.1 Summary

Overall, our project has met the client's requests to a high standard and our team worked well together to achieve this. Our documents were kept up to date and our program was built to meet all of the functional requirements. Unfortunately, there have been a few setbacks that mean that there are a few aspects of the documents and code that are not in the state that I and the client would have liked them to have been in.

2.2 Use Case Documentation

Firstly, our use case specification [1] and use case presentation [2] are true to how our program looks and behaves. Over the course of our project we have deviated slightly from the original design as we had originally misinterpreted some of the requirements, however we have attempted to update the document accordingly.

2.3 Test documentation

The test specification [3] is in a state which provides a comprehensive guide to testing our programs functionality against the functional requirements. During the project only one person worked on the test specification until the very end of the project (excluding formal review meetings). This may have contributed to us missing a couple tests for FR5 for a part of the project, however these missing tests have been accounted for and updated during system testing, and the document has now been corrected.

The test report [4] is in a good state and contains a comprehensive representation of how our program performed during system testing. We deviated from SE.QA.10 [5] where is states that: "[We should have] an explanation column (blank if passed, saying why failed otherwise)". We did in fact have an explanation column, and left it blank for the majority of the tests, but where we felt there was more context needed for the result of the test, including passes, we used the explanation column to provide that context.

2.4 Design Documentation

The design specification [6] is admittedly the least complete of our documents. While describing the general structure of the program and its classes and algorithms some of the more specific details are missing such as

aspects of the various diagramsI believe that this was originally due to a lack of clarity from myself on which parts of the design specification I expect various group members to work on; then due to a lack of motivation to correct this once we began integration of the code. However, despite it's setbacks I still believe that the design specification is a useful tool for the client to understand the inner workings of the program.

2.5 Final Program

In regards to our program it meets all of the clients requirements bar two. The two requirements could be met with a couple changes. Had we been slightly more organised and effective with our time I believe both of these requirements would have also been met and our program would be fully functional. During the course of the project we overlooked several algorithms and classes which meant that we had to rush during integration week to implement them, which was mostly successful, besides the two aforementioned issues; which were likely a result of this slight miss-management.

3. HISTORICAL ACCOUNT OF PROJECT

3.1 Introduction

During the course of the project we worked in time blocks of a week. Meeting at least once a week on a Thursday for our tutorial where we evaluated progress and assigned tasks based on skills and the current state of our project. As the project began ramping up and the complexity of deliverables and work increased we started meeting outside of the tutorials for review meetings, collaborative work sessions and progress discussions.

3.2 Week One

In the first week of our project we focused our efforts on understanding the documentation and requirements of the project, to ascertain what was expected of us as a group. We also thought about which members would be suitable for various roles and begin making basic plans for various documents and prototypes. During this first week we also all attended a practical session in which we learnt how to use gitlab and set up our repository for the project. It was crucial that we all understood this as the whole project was stored and maintained on gitlab. Two of our group members were assigned to read SE.QA.04 [7] and begin work on the use case specification, resulting in a basic plan of the use case document and a draft copy. Two other members of the group were assigned to be working on the test specification, only one of which completed work towards it; at the end of the week we had a basic list of tests for each functional requirement that we could base a document around. Two other group members began looking into JavaFX, with the intention of building their knowledge so that they could work on the on-screen display later down the line. Finally, another member began looking into the QA role and reading the documentation describing the QA process.

3.3 Week Two

The second week was when we all reported back from our research, reading and initial tasks and began formulating a structure within our team which would last for the duration of our project. The team roles were selected and by completing our tasks from last week we had established members of our group that were in a position to work on specific documents and tasks. We had also ascertained what some potential difficulties in our project would be and began thinking about how we would get around them. We also began writing crucial algorithms for the chess game such as check and checkmate. The use case document with its presentation were also worked on, as well the test specification document. A prototype was also developed, which was used in later iterations of the program for reading and writing save files.

3.4 Week Three

Week three was mostly a continuation of the previous week, but with more focus on prototypes. We now had a prototype for checking valid moves for the majority of pieces as well as more refined save-to-file prototype. The deadline for the use case document and the UI presentation were also near, so the two group members that were working on that began finalising the documents.

3.5 Week Four

By week four we had a use case document and presentation that we believed was ready for release after a review meeting and corrections were made, as well as more prototypes and code examples. We had pseudocode for a checkmate system, a prototype main menu, a somewhat refined prototype for generating all valid moves for pieces (excluding castling and en passant) and a prototype of the save-to-file system that could be integrated with the valid moves code. Plans for the design specification were also laid out (who would work on which areas etc.

3.6 Week Five

In week five there was now more focus placed on the test specification as the deadline was near and we finalised it and conducted a review meeting before release to make sure that it was up to scratch. We continued to work on the various algorithms, taking suggestions presented in the meeting. Two of our group members who were working on code prototypes began collaborating as their work had begun to clash. All members of the group were assigned to read SE.QA.05 [8] to familiarize themselves with the design document specification before working it. Once this had been done each group member began working on their assigned aspects of the document. After some work had been completed towards the document we had a meeting / collaborative work session in which we bought together our work and ideas.

3.7 Week Six

Week six was the first week in which we had feedback from the client to act upon. The feedback was for the use case document, the UI presentation and the general state of our gitlab repository. Therefore, one of the key focuses of this week was to respond to this feedback. We also continued to work on the design specification, including holding another meeting to ensure we were not overlapping on work / were on track. The QA members of our group also began researching unit testing using JUNIT and writing a report on how we are to test our code as we write it.

3.8 Week Seven

Week seven was near the deadline for the design specification so as a group we decided to prioritise finishing the document so that we had sufficient time to conduct the review process and correct any issues found. For the rest of the week we finalised the prototypes we had written in preparation of writing finalised versions of the classes.

3.9 Week Eight

In week eight we focused on writing the classes laid out in the design specification. Most group members were assigned a class to work on and where we could it would be a class that they already have experience from in

prototypes. The QA members of the team were not assigned classes to write as they were expected to look over our code and make sure that it was up to standard. They also finished researching JUNIT and produced a report on writing tests for the rest of us to refer to while unit testing. The QA also helped with writing the test spec improvements.

3.10 Week Nine

The focus for week nine was implementing the remaining methods and classes that had not been written that were essential to the basic functionality of the game. This included the JavaFX classes as well as all of the piece classes. We also had feedback from the client to respond to regarding the design specification. Due to this, tasks were assigned to those who worked on aspects of the design specification to change their respective parts to match the feedback. A formal review meeting was also conducted to ensure that this was done.

3.11 Week Ten

In week ten we began working on issues that we had found with our design and code as well as continuing to implement features into our program. Fixing issues included proposing additional aspects of our program that we overlooked, adding the additional features to the design spec. Working on en passant and castling and conducting a review meeting to ensure that the changes meet the QA standards.

3.12 Integration Week

By this point we had a functioning program but it was missing many features including special moves and replay. As well as containing many bugs in some areas. Over the weekend leading up to integration week a couple team members worked very hard to fix these issues and by Monday our program was looking much better. Despite Monday being a bank holiday all the group bar one attended a four hour collaborative work session to work on our program. The JavaFX team refined the current menus that we had in order to provide a benchmark for how they should look later on in the week as they are implemented. The castling and en passant were also now implemented.

3.12.1 Tuesday

On Tuesday the bugs we had discovered so far were fixed and the castling and en passant moves were now also added to the GUI, this meant that the players could now see that the moves were possible, before we could make the moves but it wasn't known to the player. Some team members also began writing JUNIT tests for the classes.

3.12.2 Wednesday

The next day was focused on JavaFX where we implemented the draw and resign menu, various popups and various structural changes to the program, including having JavaFX come with our package and reconfiguring where save games load to and from. Once again there were many bug fixes alongside writing JUNIT tests.

3.12.3 Thursday

On Thursday we implemented more features including, but not limited to: replay, checkmate popup, game deletion and pawn promotion. While these were being implemented more JUNIT tests were written and we started improving the JavaDoc comments for each class. We also began our rounds of system testing which helped identify missing features and bugs, which were consequently worked on.

3.12.4 Friday

On Friday we wrote the rest of the JavaDoc comments and worked on the missing JUNIT tests. We also had several bugs in the replay function to work on, the majority of which were fixed, bar one, which came back to bit us in the acceptance testing.

3.13 Final Week

In the following week leading up to the documentation deadline we updated the documents to match the final product worked on the maintenance manual and wrote the final reports. We also had acceptance testing.

4. FINAL STATE OF PROJECT

Overall, the final state of the project is mostly in line with the requirements however, some aspects have some areas for improvement.

4.1 Use Case Documentation

Firstly, the use case document is in a suitable state and matches the users and how they would interact with the program. I believe that it would be beneficial for us to have written more error conditions. This would have provided more insight into what may go wrong during development and would also be useful for the client to pass on to the maintainers so that they can understand potential issues with the program. Other this I am happy with the state of this document. Next, the use case presentation is mostly representative of the visual aspects of the game, however I believe that two of the images are outdated and no longer match the visuals in the game.

4.2 Test Documentation

I believe that the test specification is in a state that allows the client to system test our program comprehensively, it contains all of the tests necessary to detect any game breaking bugs and errors. Following feedback from the client we made some of the tests more specific, however, some more specifics could be added to do deeper testing. This would be useful for maintainers if they were to change algorithms.

4.3 Design Documentation

As mentioned before, the design document is mostly complete, with a couple outdated diagrams. These are the component diagram and the sequence diagram. Despite the outdated diagrams the document is still a useful tool for the client to refer to when they are trying understand the process of development and how our program classes and methods work together to create a functional program.

4.4 Final Program

Finally, our actual program is almost fully functional, besides a couple bugs.

4.4.1 FR10 Discussion

One of these bugs caused us to fail FR10. The bug in question was that the replay function does not work when the game is less than two moves as our code needs there to be more than two moves for the ArrayList to function correctly when extracting data from files.

4.4.2 Game Save Issue

The other requirements that we did not fully meet were FR8 and FR9 as the player is not presented with an opportunity to save the game; it is saved automatically.

4.4.3 FR11 Discussion

Another minor issue causes a slight bug with FR11, the game only restores to the previous state before the game is aborted (one move behind).

5. PERFORMANCE OF EACH TEAM MEMBER

5.1 Archie Malvern [arm36]

Archie took the role as deputy team leader and carried out his duties effectively. For the start of the project he worked on the UI and use case documents alongside Sean. Later on in the project he also worked on use case sequence diagrams for the design specification as the were related to the original use cases that he had helped write. When it was time to start writing code Archie wrote functional well structured code. During integration week, Archie wrote lots of code for the FenConverter class and its usages, as well as writing JUNIT tests for some classes. On one occasion I was ill and could not attend a tutorial, Archie filled my role as leader and covered all of the necessary topics.

In conclusion Archie was an valuable member of our team, completed all tasks assigned and filled his role well.

Blog: ..\..\config\blogs\arm36

5.2 Jasper Crabb [jac127]

Jasper took the leadership role for the group, ensuring that everything ran smoothly, a role which he carried out well. He led the tutorial meetings throughout the project, dividing up tasks amongst team members based on what they were good at. Outside of his leadership duties, Jasper also worked extensively on the test specification, and prototyped an implementation of FEN strings, the chess saving system the final program would end up using. He did a lot of administrative work as well, such as tracking hours, restructuring files in the project, setting up GitLab issues and setting up meetings. Some of these meetings involved bringing together the structure of the final system, which Jasper lead well.

In conclusion, Jasper was a consistently effective leader who did a good job of keeping the project running smoothly throughout.

Blog: ..\..\config\blogs\jac127

5.3 Jason Smith [jas160]

Jason did not complete much work or attend many meetings. This includes tutorials and our out-of-timetable meetings that we set up for various work aspects. This was apparently due to illness. He only completed 19 recorded hours of work. However, he did also show up to the last day of integration week. Due to his inconsistency attending meetings he was assigned less work than the majority of the group as we were unsure whether he would complete it. Our suspicions were often confirmed when we found ourselves playing catch up and doing extra work as work we were expecting from him wasn't complete. He was also very hard to

communicate with, often only responding to messages at night, days after they were sent. This meant that it was difficult to coordinate with him when he was assigned collaborative work. He also did not have an understanding of gitlab and did not inform us. Which meant that code that he produced did not integrate with our project and had to be modified. Jason completed some work on the FEN strings prototypes as well as working on two aspects of the design specification. One of which was not to a standard that was expected. On the final day of integration week he also wrote some JUNIT tests.

In conclusion, due to his absence and lack of contact he did not complete much work and that he did complete was not always to a high standard.

Blog: ..\..\config\blogs\jas160

5.4 Kacper Dziedzic [ktd1]

Kacper was a key asset to our team and went above and beyond to bring our project to the stage that it is now at. Kacper originally began work researching difficulties that we may face in the project. From there he developed algorithms that may be needed in our program. From there he built complete prototypes that demonstrated the algorithms in action and how they would be implemented into our program. As he had a comprehensive understanding of the algorithms he worked on the significant algorithms component of the design class. Once the design document had been written he worked hard on implementing the classes we had planned in the backend. Fixing many bugs and writing the majority of the backend classes. In the days leading up to integration week he worked exceptionally hard to bring the program to a state that could be used and worked on easily. Then during integration week Kacper implemented many backend features and fixed almost all bugs in the backend. As well as implementing some of the front end features, such as the pawn promotion and it's popup.

In conclusion, Kacper's contribution to our project cannot be understated. He was the main developer for our backend and did everything in his power to bring our project to its current state.

Blog: ..\..\config\blogs\ktd1

5.5 Muntazir Rashid [mrm19]

Muntazir was also a very important member of our team as he worked on the majority of the JavaFX and frontend, while also working on some of the backend. From the beginning of the project he worked on the JavaFX and visuals of our program. First creating a chess board, and then pieces to go along with it. Next he created the main menu and began working on the on screen piece movement. When we began writing the design specification he wrote the interface description as he was familiar with the majority of the classes and methods in the program due to him beginning to connect the front and backend of our program that same week. Next he worked on the other menus in game and added a framework that could be used to add load functionality to the game. During integration week Muntazir was responsible for many features that interacted with the frontend, including the replay board. He also worked very hard to fix bugs in our program. Alongside this he also wrote JUNIT tests for the majority of the piece classes.

In conclusion, Muntazir was our main frontend developer and was responsible for the majority of the frontend features. He worked very hard alongside Kacper during integration week to get our program to the state it is currently in.

Blog: ..\..\config\blogs\mrm19

5.6 Sean Gemmell [seg19]

Sean took the role as quality assurance manager and worked to ensure that our documents and code were met the required standard. Sean also worked on the use case documents alongside Archie. In the first week Sean discussed potential ideas for the user interface with and then began prototyping ideas. From this he was able to collaborate on the use case documents and develop an idea of how our game was going to look. He continued to update this as our ideas for the game progressed. Sean led all review meetings, except for the meetings reviewing documents that he wrote. As per the quality assurance standards (SE.QA.01 [9]). From that point

onwards we began developing code and he reviewed it as we wrote it to make sure that it was up to standard. He also researched JUNIT testing and collaborated with Wei to ensure that unit testing was carried out effectively. During integration week he conducted system testing and provided feedback to the developers based on these tests results while making sure that our project was meeting the standards.

In conclusion Sean filled his role effectively and played a big part in making sure everything was completed as well as offering to help regularly with tasks if needed.

Blog: ..\..\config\blogs\seg19

5.7 William Parry [wip24]

Will worked on both the JavaFX and backend of the code. He began by creating a JavaFX prototype of what our program would look like, which aided in implementation later in the project. Next he began working on a prototype of check and checkmate, which then used Kacper's pieces to create a prototype which functioned with a display. Although his code and algorithms were not directly used in the end product. His contributions helped with development as they provided a guide and inspiration. Will then began working on the UML class diagrams in the design specification. He then worked on fixing bugs in our program. During integration week he wrote some JUNIT tests, but most importantly helped with the system testing. As a member of the team who add worked on the code he provided some technical insight while testing.

In conclusion Will was a good team member, helping with lots of tasks and making the development of our program run smoother.

Blog: ..\..\config\blogs\wip24

5.8 Ng Wei Yuan [wyn]

Wei took the role of deputy quality assurance manager. He began by reading the QA documents and compiling a report to relay to us in the next meeting. He continued to revise document to make sure that we were following standards. In our review meetings he aided Sean in the process Wei also assisted group members in changing the documents to meet the client's feedback. He also added additional tests to the test specification that were previously missing. In integration week he oversaw the writing of JUNIT tests to assure that they were testing the classes correctly.

In conclusion Wei effectively fulfilled his role. Some of his work was not as obviously as productive as other group members, however I believe that is a result of a lack of tasks being assigned by the project leader, rather a than a lack of effort as he completed his given tasks to a fantastic standard.

Blog: ..\..\config\blogs\wyn

6. CRITICAL EVALUATION OF THE TEAM AND PROJECT

6.1 Team Performance

Overall, I believe that our team performed very well and the result of our project is something that we can be very proud of. There are a few minor bugs in the program and oversights in the documentation, but we still worked together to fulfil the clients requests to a high standard. I believe that these hiccups were caused by poor time management, both from myself (the project leader) and the other group members. Had I been more proactive in making a plan: creating a gantt chart for example and set deadlines other than those by the client many of the pieces of work would have been completed with time to spare, meaning that we could have fixed the issues that arose nearer the end of the project. At some time during the project there was not enough communication between team members, this may have bottlenecked the design process, but we eventually began collaborating and completed the design and code.

6.2 Potential Project Improvements

The project that was set for us used a waterfall lifecycle model, which is now less common in industry as opposed to more contemporary lifecycles such as agile development. If the project was set in a way that allowed choice of lifecycles or the client had asked us to use a more modern lifecycle I believe the project would have been more beneficial to us. As for the content of the project I think it provided a good range of skills for group members to work on. The only complaint that I would have is that those familiar with the game of chess had a slight advantage starting out as they would have had a greater understanding of what was required. Perhaps to combat this the project could involve a lesser known game.

6.3 Lessons Learned

For our group the most important lesson learned was definitely in time management. As the majority of us left the tasks and deliverables to the last minute, meaning we couldn't complete them to as high a quality as possible. Throughout the project we also learned the importance of communication and by the end of the project we all communicated effectively; besides one group member. In regards to lessons on software projects I believe that we have learnt to appreciate documentation and its importance, as it meant we had something to refer to at each stage of the project.

REFERENCES

[1] Use Case Document/1.2(Release)
[2] Use Case Presentation/1.1(Release)
[3] Test Specification/1.1(Release)
[4] Test Report/1.0(Release)
[5] QA Document SE.QA.10 – Producing a Final Report2.4/(Release)
[6] Design Specification/1.3(Release)
[7] QA Document SE.QA.04 – User Interface Specification Standards/1.4(Release)
[8] QA Document SE.QA.05 – Design Specification Standards/2.3(Release)

DOCUMENT HISTORY

Version	*Issue No.*	*Date*	*Changes made to document*	*Changed by*
0.1	N/A	10/05/23	N/A - original version	Jac127
1.0	N/A	11/05/2023	Release	Jac127
1.1	N/A	11/05/2023	Added Jasper's section of the "Performance of each team member" section	arm36