Introduction

Our idea is to develop a desktop application that checks computer part compatibility. Initially we will only be developing the system to focus on Intel based computers. The software will allow users to select components of a PC and will tell them if they are compatible. For example, if a user chooses a motherboard with DDR3 RAM slot they will only be allowed to choose DDR3 RAM. We chose this idea as we all have an interest in building personalised computers and have experienced issues when buying incompatible computer hardware. Therefore we are aiming this software at users who wish to build their own PCs; however the software could be used by businesses which are building PCs for their customers.

There are many constraints involved in developing a software application. One of them is time constraints, all organisation members are full time students therefore we all have other commitments. Another constraint is the current skill level of the organisation, all team-members have learnt similar programming languages in previous years. Therefore there are constraints on which languages we will be using, this is likely to be Java and MySQL as learning new languages/tools would require a lot of time spent.

Project Organisation Outline

Within the organisation each team-member has a non-technical and a technical role. This was decided by profiling each team-member and asking them what their strengths and qualities are.

Pawel is to be the Technical Coordinator, managing the group Github account and Google Drive. He will be the lead of logic and maths programming, as well as providing programming support.

Tomasz is to be the Group Coordinator, in charge of arranging meetings and providing direction as a whole. He will be the lead GUI programmer.

Lillie is to be the Project Coordinator (dubbed 'HR'), ensuring milestones are met and chasing up any incomplete work. She will be working on the main application and assisting Tom with creating the database.

Luke and Greg are to work together checking all documentation and providing charts/visualisations. Both will work on programming the main application, and Luke will assist in GUI programming.

Tom is to be the Secretary, writing up minutes/notes of the weekly meetings and ensuring all documentation is assembled ready for hand-in. He is the most comfortable with databases so will be leading the database design.

Risk Analysis

The organisation has conducted a risk analysis for the system development, it highlights all possible risks, their likelihood, their effect, prevention methods and counter action. The full risk analysis is located at *appendix 1*.

Resource Requirements

This project will require many resources for a successful completion; these are detailed below.

- Personal resources - Time (Organisation members are full students), Skills (Each member has a different skill level, as stated in the project organisation section)
- Hardware - Organisation - Members will use their personal computers to complete work as well as the university's computers. Plus a VM will be used for a server.
- Software - The organisation will be using a variety of software to develop the software. We are using NetBeans for the Java development, this is both for logic and GUI. We are using a VM server to host the MySQL database. Additionally, we will be using Notepad++/Atom to store MySQL code.
- Programming Languages - Java and MySQL, with the possibility of HTML5 & CSS for web-end development.
- Financial- The organisation requires no capital to complete this project.
- Location - The organisation is using the library as a group meeting space every Wednesday 1pm. Additionally, organisation members are working from home and using Facebook Messenger to communicate with the rest of the group.

Work Breakdown and Milestones

The organisation has determined the most important deadlines which need to be met throughout the duration of the project. These milestones will help to keep the organisation stay on track, and determine whether the project is on schedule, and if a milestone is missed, can be addressed immediately and determine an extended deadline without breaking schedule of the rest of the project.

Work Breakdown/Milestones	Deadline
Project Proposal & Plan	14th Oct 2016
System Requirements Specification	18th Nov 2016
Design Documentation	23rd Dec 2016
Database Design Completion	30th Dec 2016
Application Design Completion (Logic)	30th Dec 2016
GUI completion	9th Feb 2017

Debugging Application	17th Feb 2017
Prototype & Demo Video	17th Feb 2017
Testing Documentation	3rd March 2017
Final Report & Application	28th April 2017

Most deadlines here are simply the dates for the hand-ins of each section, so as much as we'll be handing in work on those dates, things will be tweaked as we continue the project, especially when creating the database and main application.

Project Schedule

To schedule the project efficiently we created a Gantt chart (*see appendix 2*) to help us visualise our timeline more clearly. Many of our deadlines are based around the coursework hand-in dates for each submission. We tried to give the larger tasks (such as application programming) more time to complete as complexities are expected. We delegated tasks between ourselves based on individual ability and also preference. Some tasks are dependent on the completion of others, but the whole nature of the project and our hand-in deadlines forces things to be fairly linear in regards to what order we complete things anyway so this isn't a problem.

Monitoring Mechanisms

Throughout the task different sections of the project will have to be completed in different time scales, to be able to meet deadlines, team will have to be well organised. Each week there will be at least one meeting with the whole organisation, each organisation member will receive a task to complete by a deadline, set by the organisation members which will be several days before the official deadline. We will also work towards finishing a section days before it is due to have time for any changes that might come up.

Each organisation member will need a clear understanding of what the outcome will look like, so that everyone is working in the same direction. Also everyone in the organisation needs to put the same amount of effort and delegate time so that we are able to complete the task on time.

On each meeting one person will make minutes in order to keep track of what we have already done and what is the plane for next week, those minutes will include:

- Attendance
- Work complete
- Plan for next week
- Tasks for each team member
- Any changes to plan
- Any problems that occurred

This way our work will always be organised and easy to follow as everyone will be able to access this and have a read to check what they need to do. All of the work will be stored on a Shared Google Drive so that each team member has access.

Appendix 1: Risk Analysis

Possible Risks	Likelihood	Effects	Prevention	Counter-Action
Estimation				
A. The time estimated to develop the software is underestimated.	Unlikely	Catastrophic	Ensure that the software requirements are not too challenging before development starts.	Change requirements of the software to suit the time frame. Allow for evolution of the software in the future.
B. The complexity of the software is underestimated.	Likely	Serious		
C. The size of the software is underestimated	Unlikely	Serious		
Organisational				
D. The organisation undertakes a small reshuffle; where two people switch roles and tasks.	Likely	Moderate/ Low	Ensure roles are given to members whom have the skills and are happy to complete the task.	Brief team-members on their new roles.
E. The overall structure of the organisation is changed. So that different management is responsible for the project.	Unlikely	Serious/ Moderate		Brief team-members on their new roles. Ensure that new management is heading in a direction agreed by all members.
F. The organisation run out of financial capital to continue the development.	Very Unlikely	Catastrophic	Ensure that only free-software, or software accessible at university, is used.	Speak to Dr Claudia Iacob about possible capital for licence(s)
People				

	Likelihood	Severity	Mitigation	Contingency
G. Organisation member is ill/absent during a phase of development.	Likely	Moderate	Ensure team-members are briefed on each others role in case of absence.	For short term absence assess the effect on the long term development. If low, team-member can telework or leave work until they have returned. If high, task is given to other member(s) to complete.
H. Organisation member is (seriously)ill/absent during (critical) stages of development.	Unlikely	Moderate/ Serious		Work is given to another team-member(s) to complete.
I. The organisation lacks the skills to complete the development of the software.	Unlikely	Catastrophic	Ensure that team-members have skills in their given task before starting development.	Team-members can go on online training courses, for example Code Academy.
J. A team member leaves the organisation during development.	Very Unlikely	Serious	Ensure team-members are briefed on each member's role and responsibility	A possible reshuffle of the organisation may be needed. See risk D/E.
K. Communication method (Facebook Messenger) fails during critical stages of development.	Very Unlikely	Catastrophic	Mobile phone numbers to be exchanged between members.	Team-members to use mobile phone communications (SMS & WhatsApp)
L. Team-member fails to turn up to meetings.	Likely	Serious	Remind all team-members of meetings in group chat 5	If persistent contact Dr Claudia Iacob to discuss their role

			days before meeting. Then again on the day.	within the organisation.
Requirements				
M. The user(s) propose a new set of requirements during development.	Unlikely	Serious	Before development, ensure users are happy with the set requirements.	Discuss with users about compromising some requirements.
N. Limitations to the system are not accepted by the user(s).	Unlikely	Serious		
O. Compatibility check between hardware is not developed.	Unlikely	Serious	Before development limit check for certain hardware.	Continue with development leaving scope for compatibility check in future deployments.
Technology				
P. All work saved on GitHub is deleted/missing.	Unlikely	Catastrophic	Ensure that team-members keep back ups of their work.	Restore GitHub to last available point. Team-members to upload their work.
Q. All work saved on Google Drive is deleted/missing.	Unlikely	Catastrophic	Ensure that team-members keep back ups of their work. Secretary to download all files on the searched Google Drive folder weekly.	Secretary to reupload last weekly backup. Team-members to upload latest work.
R. The programming languages chosen are incompatible.	Unlikely	Serious	Extensive research into programming languages and	Change programming language(s) to ones which are

		8	their compatibility before starting development.	compatible.
S. The database fails to store all of the required data for the software.	Very Unlikely	Serious	Ensure that the database can handle large amounts of data before starting development.	Change language or media storage of the database.
T. The server containing the database fails, all data is lost.	Probable	Serious	Ensure all MySQL code is kept.	Ensure that the server is maintainable, then re-enter the MySql code.
U. The server is unable to connect to the developed software.	Unlikely	Catastrophic	Test connection before starting development.	Change to a local stored database on the device running the software.
Tools				
V. Software development tools are unable to support the development of the software.	Unlikely	Catastrophic		

Appendix 2: Gantt Chart

#	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Member Assigned
1		Project Start	0 days	Wed 28/09/16	Wed 28/09/16		
2	✓	◢ Project Proposal & Plan	12 days	Wed 28/09/16	Thu 13/10/16		
3	✓	Preliminary Analysis	7 days	Wed 28/09/16	Thu 06/10/16		All
4	✓	Propose Alternative Ideas	7 days	Wed 28/09/16	Thu 06/10/16		All
5	✓	Evaluate Pros/Cons	7 days	Wed 28/09/16	Thu 06/10/16	4	All
6	✓	Scrutiny of Existing Systems	7 days	Wed 28/09/16	Thu 06/10/16		All
7	✓	Team Organisation	7 days	Wed 05/10/16	Thu 13/10/16		Tomasz,Lillie,Tom
8	✓	Breakdown Tasks	7 days	Wed 05/10/16	Thu 13/10/16	7	Luke,Tomasz
9	✓	Risk Analysis	7 days	Wed 05/10/16	Thu 13/10/16		Tom
10	✓	Identify Resource Requirements	7 days	Wed 05/10/16	Thu 13/10/16		Pawel
11	✓	Propose Monitoring Mechanisms	7 days	Wed 05/10/16	Thu 13/10/16		Greg
12	✓	Produce Project Schedule	7 days	Wed 05/10/16	Thu 13/10/16		Luke
13		Project Proposal & Plan Deadline	0 days	Fri 14/10/16	Fri 14/10/16		
14		◢ System Requirements Specification	26 days	Fri 14/10/16	Fri 18/11/16		
15		Define System Requirements	14 days	Fri 14/10/16	Wed 02/11/16		All
16		System Analysis & Description	14 days	Fri 14/10/16	Wed 02/11/16		All
17		User Feedback & Requirements	14 days	Tue 01/11/16	Fri 18/11/16		Lillie,All
18		Plan Out Documentation	7 days	Thu 10/11/16	Fri 18/11/16		All
19		System Requirements Specification Deadline	0 days	Fri 18/11/16	Fri 18/11/16		
20		◢ Design Documentation	26 days	Fri 18/11/16	Fri 23/12/16		
21		Use Case Models	7 days	Fri 18/11/16	Mon 28/11/16		Tom,All
22		Sequence Diagram Models	7 days	Mon 28/11/16	Tue 06/12/16		Tom,All
23		System Architecture Models	7 days	Tue 06/12/16	Wed 14/12/16		Tom,All
24		UI Mockups	14 days	Tue 06/12/16	Fri 23/12/16		Luke,Greg
25		Design Documentation Deadline	0 days	Fri 13/01/17	Fri 13/01/17		
26		◢ Prototype & Demo Video	41 days	Fri 23/12/16	Fri 17/02/17		
27		Java Programming	35 days	Fri 23/12/16	Thu 09/02/17		Tomasz,All
28		Database Programming	35 days	Fri 23/12/16	Thu 09/02/17		Tom,Lillie
29		Algorithms, Efficiency & Debugging	21 days	Sun 22/01/17	Fri 17/02/17		All
30		Initial Build Demo Video	14 days	Tue 31/01/17	Fri 17/02/17		All
31		Prototype & Demo Video Deadline	0 days	Fri 17/02/17	Fri 17/02/17		
32		◢ Testing Documentation	11 days	Fri 17/02/17	Fri 03/03/17		
33		Testing Strategy Analysis	7 days	Fri 17/02/17	Mon 27/02/17		All
34		Testing Process Details	7 days	Fri 17/02/17	Mon 27/02/17		All
35		Testing Implementation	11 days	Fri 17/02/17	Fri 03/03/17		All
36		Testing Results	7 days	Thu 23/02/17	Fri 03/03/17		All
37		Testing Documentation Deadline	0 days	Fri 03/03/17	Fri 03/03/17		
38		◢ Final report & Application	36 days	Fri 03/03/17	Fri 21/04/17		
39		Usability Evaluation/Write-Up	21 days	Fri 03/03/17	Fri 31/03/17		All
40		Final Evaluation Write-Up	21 days	Fri 24/03/17	Fri 21/04/17		All
41		Final Program Demo Video	14 days	Tue 04/04/17	Fri 21/04/17		All
42		Compile System Documentation	14 days	Tue 04/04/17	Fri 21/04/17		All
43		Final report & Application Deadline	0 days	Fri 28/04/17	Fri 28/04/17		