Introduction

The majority of university students live away from home, in student accommodation or rented our houses therefore causing many problems. This could include paying bills fairly, working at the right surroundings and security. To put these into examples, why is the electricity bill so much and who is using the most of it? Items may go missing and doors may be left unlocked or even opened by someone else. Also a student who is trying to work is struggling, due to the room being not right, for example at the correct temperature. Therefore a system which can tackle these problems would make a student's life easier in many prospects.

The concept behind this project is to develop a sensor-based system operating on the Edge which collects and analyses data from shared student accommodation. Data will be anonymously collected and analysed in real time in order to provide insight into student accommodation and to improve shared living arrangements. In order to do this, we aim to implement a sensor-based system to be installed in student flats, and to provide a web-based service in order to access certain aspects of collected data. While this will primarily be geared towards students, it can easily be marketed towards third party organisations such as rental agencies and landlords.

The team has established the following possible constraints:

* Operating systems and platforms supported: the team must ensure that the system works on University machines (primarily Windows, OSX, and Linux) in order to meet stakeholders concerns.
* The use of a specific library or framework (IBM’s software)
* Programming language: the team has decided that we all must code in one programming language to make component integration an easier process.
* Schedule: there is a project deadline that must be followed.
* Budget: the budget constraint is primarily based on the cost of the sensors. We aim to get cheap sensors that provide the functionality we require.
* Team composition: we have come to recognise that we all have different, busy schedules. Stakeholders at IBM, Wendy, and the team all have different schedules and we must work around them.
* Project environment and targeted clients: students can be unpredictable, which may lead to skewed data. Certain students may think this project is an invasion of privacy and tenants may disagree amongst each other.

The software development model that the team has decided on is called Rapid Application Development model (RAD Model). This will allow us to develop the components in parallel as individual mini projects. The development of the components will occur in a time frame and will then be delivered and integrated to produce a working prototype. Furthermore, this will allow us to quickly give our client something to see and use and to provide feedback regarding the delivery and their requirements. Additionally, this approach will allow us to constantly integrate our components from the beginning reducing integration issues. Another reason we picked this approach is because we have a small period of time for the development phase and this reduces the development time need to produce a prototype.

Risk Assessment

|  |  |  |
| --- | --- | --- |
| Category | Potential Risk | Risk Management Technique |
| Timing | Underestimating time required to complete the project. | * Follow the Learning Central deadlines * Set and meet additional deadlines * Workflow diagrams |
|  | Unable to plan group meetings when everyone is available. | * Doodle Poll * Identify when everyone is available * Follow the Gantt Chart |
| Requirements | Client could request to change requirements at any point of the project. | * Ensure constant communication with the client at all stages of development giving them the opportunity to view progress and provide feedback * If requirements are changed, organise a group meeting and discuss what further tasks need to be done by the team. |
| Uncompleted work | Work by a team member may not be complete due to certain circumstances. | * Ensure workload is spread evenly. * If failure to complete the work is anticipated then we would be able to delegate the remaining tasks to other team members. * We can also hold progress meetings to ensure that we are not falling behind on a particular task. |
| Human Resources | All decisions may not be supported by all group members. | * Use a voting system * Rational decisions based on the majority of member’s thoughts |
|  | Delaying the group project because members of the team are absent. | * Facebook group – members can keep in touch and notify others of any absence * Google Drive – members can share and store work in same location * Whatsapp * Doodle Poll |
|  | Certain functions may not be implemented as members of the team do not have the skills and requirements to do so. | * Delegate work to members who feel more competent of completing the given task * Work and coach each other |
| Productivity | Motivation may decrease, as the timeline of the project is long. | * Set a time frame for each task to be completed in * Each group member has one or more tasks to do * Each member should be motivated to complete set tasks |
| Operational | Functionality and time loss due to hardware failure. | * Address priority conflicts in regards to the software * Split responsibilities * Weekly group meetings to address any unexpected issues |
| Technical | Lack of thorough system testing could lead to flaws and bugs. Hence some requirements may not be fulfilled. | * Conduct a full test on every function of the system * Test system thoroughly against acceptance criteria |
|  | Software could potentially not fulfill system requirements/may not work at all . | * Ensure system can be implemented correctly * Establish final requirements as soon as possible to focus on design and implementation |

Legal Issues

**Liability**: Users will need to accept a disclaimer that no warranty comes with the system. Any damages incurred will be the responsibility of the user.

**Security**: Data is collected and stored anonymously and in a secured environment, most likely to be IBM’s “Edge”.

**Possession of data:** Users must know who is possessing their data, and be informed somewhere if their data is being accessed by a third party.

Social Issues

**Invasion of privacy**: We must ensure that users are not experiencing an invasion of privacy and that sensors are collecting anonymous data.

Ethical Issues

**Ability to opt-out:** It will be difficult for users to opt-out once sensors have been set up and installed. Seeing as how the sensors belong in a shared living environment, this provides an issue should one person want to revoke their involvement.

Effectiveness of communication and suggested methods:

Our methods of communication are as follows:

* WhatsApp
* Facebook
* Google Drive
* Slack
* Group meetings
* Skype
* Doodle Poll

WhatsApp and Facebook allow the group to communicate frequently in regards to planning meetings or clarifying minor queries. Google Drive allows the group to share their documents and work on the same document at the same time. Slack eliminates the need for email. It allows the team to constantly communicate, along with its other integrations. This will allow the team to integrate Github as well as our Google Drive documents. Our Skype meetings really just occur when the team cannot meet in person. Therefore Skype becomes a good replacement by allowing the team to communicate even when apart. Group meetings are a necessity as this is where the group sits down for hours at a time constantly making decisions and allowing us to work as a group. Doodle Poll is just used to plan our meetings around our busy schedules. The wide variety of ways we are going to communicate reduces any risks of communication problems.

Work Plan

1. Meet as a group
2. Brainstorm Ideas
3. Individual Research
4. Present Ideas to Wendy and IBM
5. Refine Individual Research
6. Meet as a group and discuss
7. Present Refined Ideas to Wendy and IBM
8. Meet as a group and discuss:
   1. Constraints
   2. Social,legal and ethical issues
   3. Risk assessment
   4. Scope and Boundaries of the system
   5. software development model
9. Write up part of the report
10. Present to wendy
11. Finalise project
12. Complete report
13. Initial Design steps
14. Co-Design /Participatory Design
15. Mock Ups
16. Develop Technical Probes
17. Deploy Technical Probes
18. Testing
19. Integrate to create system
20. Testing
21. Evaluate
22. Present the system

Gantt Chart

Deliverables

|  |  |
| --- | --- |
| **Deliverable** | **Description** |
| Functional/Non-functional requirements | Full list of system requirements. |
| Acceptance criteria | Testable criteria for all requirements. |
| Social, Legal & Ethical issues | Document discussing issues regarding our system. |
| Risk Assessment | Identify problems and implement solutions. |
| Scope & Boundaries of system | Document identifying the overall scope of the system. |
| Component Diagram | How components are wired together to form larger components and or software systems. |
| Gantt Chart | Bar chart that illustrates a project schedule. |

|  |  |
| --- | --- |
| Development Model | Discussion of most suitable development model to use. |
| UML Diagram | Addition of attributes to describe a range of instances of a class. |
| Class Diagram | Set of objects to show attributes, operations and relationships. |
| Use-case Diagram | Illustrates relationship between actor and use case. |
| Use-cases | Describes the functional requirements. |
| Mock ups | Wireframe mock up of system. |
| Evaluation | Evaluate the overall system created. |
| Programming language | Justification for programming language to be used |

Acceptance criteria

These acceptance criteria can be used during the testing phases to provide tangible results to indicate whether or not the system has met the requirements. We have tried to quantify as many of the acceptance criteria as possible to allow for accurate comparison of the test results. It is assumed during these acceptance criteria that the user has “(data to analyse? internet connection to analyse? Sensor?)“. **Need to know further info**

Development model - Agile

The model we have chosen to use is Agile method. Agile allows changes to be made to the project and initial planning, and for this project client changes are expected. Also as we will have 4 individual components to which each team member will have to write, re-writes the program will be helpful as everyone's code will be combined together at the end. The technology industry is always changing and upgrading, agile methodology allows changes to be made therefore features will be easier to add. each week we will set ourselves tasks and meet, followed by a meeting with the client every fortnight. With the client adding to the feedback, any changes or comments will make our system what they desire. On the programming stage, the tests at the end of each week that our work is set will allow any bugs to be caught and solved. Therefore the end product will not be faulty and run smoothly.

Why not Waterfall?

The reason we didn't choose waterfall is because changes cannot be made to the previous steps complete, and with this project client changes and expected.