Student Housing Group Project

Project plan

Date: 26.04.2023

Group: S2-4

1. Introduction
2. Project Definition
   1. Project Background
   2. Project Definition
   3. Project Goal
   4. Expected Result
   5. Way of Working
   6. Scope
3. Project Structure Organization
   1. Project Leader
   2. Secretary
   3. Chairman
   4. Project Member
   5. Tutor
4. Risk Assessment
5. Functionality
6. Deliverables
7. Priority
8. Planning

1. **Introduction**

We are group S2-4 (Danila S., Angel R., Ivan E., Ivaylo B.). We are making an IT solution for student housing project from week 10 to week 18. In this project we will focus on implementing concepts we will learn during these 8 weeks. If we have enough time will try to push beyond basic requirements.

1. **Project Definition**

2.1 Project Background

The main issue that should be solved by this project is students’ irresponsibility towards common places such as kitchen and restrooms. This happens when it is difficult to find rules of the place and there is no communication about the schedule among residents.

Current problems with the dorm:

* Appointed persons not cleaning the shared facilities.
* Groceries are not done or paid for shared items such as toilet paper, dish soap, etc.
* Garbage disposal is not done on time.
* Unannounced parties, gatherings, etc.

2.2 Project Definition

Stakeholders are looking for a solution to the problems they stumbled into (the problems are listed above).

House owners find IT solutions the best possible tool to solve their problems. They are open to all possible ideas that can help them and turn the dorm into a cleaner, more peaceful and better place.

2.3 Project Goal

The main goal of this project is to build up students’ motivation for doing their house chores like cleaning and taking out trash, also make it possible for residents to communicate properly about parties so that they don’t disturb anybody. The motivation system is going to work on awards and not punishments as the first on is much more efficient judging by various psychological researches. If the student does all the stuff about the house he should do he will get a discount for his house rent which will raise his motivation to clean common places.

2.4 Expected Result

We are planning to get a working software with a pleasing interface for students. It will contain all the rules of the apartment, schedule for taking out trash, cleaning and party hosting. And a voting system for parties. It will be based on windows forms.

2.5 Way of Working

The work between all the teem members will be distributed equally. Everyone is going one part of the project (for example one will be creating form for taking out trash another one will be working on party arrangements and so on).

All of us have similar skills in programming that is why we will be able to deliver product of the same quality, but to avoid mistakes everybody will be able to leave feedback on someone’s else work. Such comments shouldn’t be aggressive as their only aim is to figure out the problem of some part of code.

Our workflow will be based on deadlines without many planed meetings. After all the discussion about the structure of the program we will start working with our own temp. If there are any problems that occur with someone’s part our main priority is to help that person and continue working so that all the parts would be ready on time. On Wednesday each week we will have small discussions to be clear that everyone is doing well with their part. These meetings are compulsory to visit and missing them 2 times without any reason will cause in getting thrown out of the team.

2.6 Scope

The scope for the project is to deliver a working windows application for students using C#. Our program will have a windows for authorization, house chores and rules. Also we will display the information about rewards.

We might add a server part to the project to make google authentication work. But that task is an extra feature and it is out of scope for now.

1. **Project Structure Organization**

3.1 Project Leader – Danila

* Structure the workflow of the team (assigning all the tasks equally).
* Being aware of deadlines and telling to other team members.
* Conduct meatings and gather the group members.
* Be a part of the team working on the project and motivating others.
* Help with the task if something is not clear.

3.2 Secretary – all

* Make notes during group meetings.
* Register for meetings.

3.3 Chairman – all

* Send agenda for the meeting.
* Follow the time and lead the discussion properly.

3.4 Project Members – Angel, Ivan, Ivaylo, Danila

* Every team member should responsible with the work he has.
* If somebody stumbles into problem he must immidiately ask for help to find the solution as quickly as possible so that the work doesn’t stop.
* Responsible with showing the work to all the team members and ready to explain the code.
* Push tested changes to the repository, not raw code that has a lot of bags in it.

3.5 Tutor

* Provide help whan a problem accurs.
* Give feedback, tips and pieces of advice that can improve code and structure.
* When any problem occurs or any part that project group that stuck, he/she needs to show the way about how to solve it.
* Review all the deliverables from the project group.

1. **Risk Assessment**

This part is essential to make agreements on some aspects of the workflow for the project. There might be unpredictable events and it is good to have solutions for them beforehand. This part includes analysis of risks that might occur and influence our work. The probability is ranged subjectively using this scale: highly unlikely – unlikely – likely – highly likely. For impact we used the scale: slightly harmful – harmful – extremely harmful.

1. **Misunderstanding in the team about the requirements.**

Misunderstanding the requirements might be the worst issue in IT world and this project is not an exception. It might lead to rewriting huge parts of code and huge overwork for all members of the team.

Solution: detailed discussion of the structure of the program, classes, methods and design.

|  |  |
| --- | --- |
| Probability | Impact |
| Unlikely | Extremely harmful |

1. **Stakeholders have inaccurate expectations.**

Stakeholders might not know what they want from the final product themselves or are not aware of the difficulty of some changes in the code, so they can ask to change all kind of stuff throughout the working process which can lead to a lot of extra work.

Solution: make agreements with stakeholders about all the basics of the project. If they want a change later, they will have to pay for that.

|  |  |
| --- | --- |
| Probability | Impact |
| Likely | Extremely harmful |

1. **Bad organization of work.**

Hundreds of projects fail to deliver good result because of bad work organization. It can make the workflow very confusing and difficult.

Solution: discuss the workflow with the team members and be able to separate all the duties equally (taking abilities of the group into account).

|  |  |
| --- | --- |
| Probability | Impact |
| Likely | Harmful |

1. **Failure to follow the pace of the project.**

Everybody should work with the same speed to avoid situation where some parts of the project are not done by the final week, or one person ran to far forward and everybody else can’t understand his work.

Solution: conduct meetings to discuss all the work that has been done so far so that all the members of the team know what the next steps are and what has already been done, in this way it becomes clear for everyone how much work they should do not to fall behind or run too far away.

|  |  |
| --- | --- |
| Probability | Impact |
| Highly unlikely | Slightly harmful |

1. **Lack of knowledge.**

It might happen that our team will run into a problem that we wouldn’t be able to solve. That can lead to a complete stop of the workflow.

Solution: immediately contact tutor to figure out the problem and try to come to a solution together.

|  |  |
| --- | --- |
| Probability | Impact |
| Likely | Harmful |

1. **Functionality**

When a student enters the app, he will have to create an account or log in if he already has one. When logged in student will be shown all the rules of the dorm. After that he will be redirected to a calendar, where he can see tasks which he and his housemates must complete on certain dates. Students will be able to mark a task as completed and report other students if they haven’t done their chores.

Administrators will have a different page. They will also be able to see all the tasks for students and they will have information on how many times was certain student reported and for what. Administrators can assign new student to the household, manage house rules and change tasks on the calendar.

1. **Deliverables**

* **Project plan:**

The overall documentation containing all business information about the project. It should have all the information we got to create this project in a clear and neat way.

* **Prototype:**

A prototype of the project. It should show the stakeholders how the application will look like when it’s done and make it clear for us how to build it.

* **Frontend:**

A complete version of the prototype without any functionality inside. It might be done with C# WPF.

* **Backend:**

Implementing all the functionality to frontend.

* **Final presentation:**

Presenting our final functioning product and explaining all the features and work that has been done.

1. **Priority**

Must have:

* Encapsulated communication between classes.
* Bug free implementation.
* Databases to store information about users and events.
* Checking everybody’s work and code during the project.

Should have:

* Google authentication.
* User friendly interface.
* Pointing system per user.

Could have:

* WPF platform to work in.
* Party planning for one specific day.
* Comments for all the events in the calendar
* Add decline button with automatic resigning.

Won’t have:

* Won’t have a website.
* No rent discounts.
* No punishment system.

1. **Planning**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project Group - 2.4 | W  10 | W11 | W 12 | W13 | W14 | W15 | W 16 | W 17 | W18 |
| Project set-up. Start Project Plan. Brainstorm features. |  |  |  |  |  |  |  |  |  |
| Project Plan. User interface design. |  |  |  |  |  |  |  |  |  |
| User interface implementation. UML class design diagram. |  |  |  |  |  |  |  |  |  |
| Continue implementation. Start implementation of information persistence in files. |  |  |  |  |  |  |  |  |  |
| Continue implementation. |  |  |  |  |  |  |  |  |  |
| Continue/finish implementation. |  |  |  |  |  |  |  |  |  |
| Finish up all deliverables. |  |  |  |  |  |  |  |  |  |
| Final presentation. |  |  |  |  |  |  |  |  |  |