A software solution for Zoo Bazaar

Student Reader (Waterfall)

7-7-2021

Version 2022.1 (Februari 2022)

TABLE OF CONTENTS

1. INTRODUCTION

As a second semester Software student you have acquired the basic knowledge of programming, databases and other IT related subjects. A question you might have asked yourself, is how they relate to each other. Serving as the thread that connects this semester together, this project will provide an answer to this question. Therefore, you are expected to invest a considerable amount of time into this project. Like the project from the advanced part of the first semester, you will find that a total of two days per week are dedicated to the project.

During this project, you will work six weeks on a software solution for a zoo that is establishing itself in the Netherlands. By applying the waterfall methodology, a final product should be created. This product will be presented to the client at the end of the six weeks. Bear in mind that your final product needs to satisfy the requirements of your stakeholder (the client). If you complete this project, the client will likely want to extend your final product with additional features for another 12 weeks!

The company Zoo Bazaar (daughter company of Jupiter) is planning to open their very first zoo in Eindhoven. Because the administration is the backbone that manages all aspects of the company, management would like to hire a team of Software developers to implement an administrative system.

In this document you will find a more elaborate description in **about the assignment**. Following, you will find the educational **outcomes** for this project, details about the **assessment** of your work, the **way of working**, details **about the assignment**, the **deliverables** and an indication of a **schedule**.

2. OUTCOMES

After successfully finishing the first six weeks of this project you will be able to:

- Successfully collaborate with team members.
- Extract requirements from text and communication with a client.
- Plan and execute a software project by applying the Waterfall methodology.
- Execute the project in a professional manner. (communication, meetings, etc.).
- Scope the project to a realistic scale.

For details on how these objectives are assessed you may refer to **assessment** and **assessment sheet**.

3. ASSESSMENT

In this project you work together as a team on a product. Furthermore, you work for a client under supervision of a tutor. As suggested in the **outcomes**, a project contains various challenges. Next to the technical challenges, you will find that collaboration, planning and communication will also require your attention. Make sure that you communicate clearly with your tutor and client, and that you are respectful to your team members. Be on time, communicate when you cannot make it to a meeting or deadlines, etc.

Two teachers will be involved with your project. One teacher will be assigned as your tutor, they will act as a senior colleague/manager, observing your process and guiding you through roadblocks when required. The tutor will assess your group on your process, documentation and deliverables looking also at the technical side of your products.

The second teacher will be assigned as your client, they will observe you in the few times you meet each other and should not be aware of any of the technical aspects. The client will assess your group on how you conduct meetings, if you ask the right questions and if you deliver what you promised. They will assess your product on a shallow level (what you present and possibly the user experience).

You may find more information about how the client and tutor will assess you in the **assessment sheet**.

Although your assessment will likely conclude in the same result as your group members, this project is assessed on an individual basis. Hence, it is possible that your tutor and client see reason to grade you individually.

It is important to know that the waterfall project (first 6 weeks) is part of a bigger project and acts as one out of several “datapoints” for your tutor and client to assess you on.

4. WAY OF WORKING

During this project you will apply the "Waterfall" methodology for managing your project. This requires you to initially design your solution and then implement it. For this reason, you will be expected to create several documents: Project plan (definition of your project), URS Document (design of your product), see **project plan and user requirements specification structure** about the structure of the documents. Furthermore, in **schedule** you may find an overview of the project schedule.

4.1. ORGANIZATION

The project is carried out in a project team consisting of three students. Presence at the classes and meetings is mandatory. Missing two or more required attendances will have severe consequences on the assessment of your professionalism.

4.1.1. CREATING GROUPS

A group consists of four students, if there is a reason for a different number of members, the teachers will decide how this is allocated. Additionally, a group may contain at most one student that has to retake the semester. On the canvas page, register your group for the project. Once you did that, you can start by preparing for the first meeting. Once your group is complete, a tutor will be assigned to your group and inform you about when the first meeting takes place.

If you are delayed, hopefully, you have contacted your teachers (and mentor) about your delay. If not, you are recommended to do so first. Second, contact the semester coordinator to make a plan of approach.

4.1.2. MEETINGS

Every week you meet your tutor twice, a weekly startup meeting and a weekly update meeting. The first meeting will be scheduled by your tutor, you are expected to schedule all other meetings. The startup meeting is intended to review the bigger picture, are you still on track and what are you planning to do this week? The update meeting is about your weekly progress, what did you do so far, are there challenges that require attention/discussion, and will you finish this week? Besides these points, you can ask for feedback on your work such as the documentation or implementation.

4.2. VERSION CONTROL

During this project you are expected to create and use a GIT (lab) repository. Most importantly this gives your group the ability to collaborate simultaneously. It has additional benefits of allowing mistakes to be undone and much more. For the tutor it is also beneficial because it gives them an overview of who contributed to the project. It is therefore not the intension that you always work together on one computer, but also work individually on the project.

5. ABOUT THE ASSIGNMENT

As mentioned in the **introduction**, “Zoo Bazaar” is opening their first zoo in Eindhoven. Funded by the parent company “Jupiter”, they intend to start as well-prepared as possible. The biggest challenge that Zoo Bazaar’s management foresees, is keeping track of their employees and animals. They want to hire a team of software professionals to develop an administrative system that allows its users to keep track of the resources (e.g. employees animals, locations, etc.). Furthermore, it should be possible to allow the users to create a feeding timetable for the animals. The system could possibly become bigger in the future. Ideas such as the addition of (online) ticket sales, statistics, other types of timetables, customer complaint handling and a dedicated website for employees to view information about their schedule can possibly be added in future versions. But for now, they would like to focus on the management of employees and animals.

A transcript has been made from the **initial interview** between your team’s representative (James) and one of the managers of Zoo Bazaar (Henriette). This will give you a start to create an initial version of the project plan. Use this interview as input for additional questions you might have.

6. DELIVERABLES

For this project you will have to deliver several artifacts at various moments. The following artifacts are at least expected.

6.1. DOCUMENTATION

- Agenda's and minutes of every meeting
- A **project plan**
- A **URS**
- A **process report**
- A test plan for your own project
- Test report for the other group's system

6.2. SOFTWARE

- An application according to the client's requirements

7. SCHEDULE

This project is split in two parts. You will start off with the waterfall project management approach in which you plan, design, build, and present your product. The next part will take a different project management approach which you will receive more information about in week seven. Below you find a table with a rough schedule.

	Waterfall					
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Deliverables/ Actions	Project plan	URS	Implementation		Test plan and binaries to peers Implementation	Software Presentation

7.1. WATERFALL

Week 1:

- Create a name and logo for your group
- Interview client
- Create and hand in project plan

Week 2:

- Improve project plan based on feedback
- Create URS

Week 3:

- Improve URS based on feedback
- Implement software solution

Week 4:

- Implement software solution

Week 5:

- Implement software solution
- Create test plan (workshop)
- Send test plan and binaries to peers
- Review peers and create test report (fill received test plan in).

Week 6:

- Tag software as deliverable in GIT repository
- Present final version to client and peers

7.2. BEYOND?

As you should know by now, the client has big plans for this system, they did not hide that there is more work to be done. If you “play your cards right” you might be asked by the client to continue with your product!

APPENDIX A: INITIAL INTERVIEW

Person	Said
James	Hello Henriette, thanks for taking the time for this interview. I would like to start by getting an idea on what you would like the system to eventually grow into. What is your vision for a possible final version?
Henriette	Hello James, we do have some ideas of what we want to achieve, but they are still subject to change. We envision a system in the end that can manage various aspects of our zoo. A final version should allow our administration to handle everything from employee administration including attendance and scheduling, to manage our animals including tasks related to their welfare. It would be great if we can also handle our front-office tasks, such as, (online) ticket sales/booth, customer service, performance statistics, etc. For our employees, we would also like to have a website on which they can log in to view and adjust their personal information and view information about their work shifts. I can probably continue with more features, but this is the rough idea for now.
James	Thanks Henriette. Indeed, this gives an idea of how you envision your system to end up looking. Let's zoom in on the scope of this project, where are your priorities?
Henriette	Currently, we have a lot of trouble managing employees and the animal's care, so our priority is on these points. If I must prioritize one above the other, I would prioritize our animals as managing their welfare is more complex compared our employee management. All other aspects can be implemented in the future, but this is the most important.
James	Clear, priority on the animal with their welfare and then employees. You mentioned management and on a few occasions, but it might actually be good to get aligned on what this means. Can you explain what information you want to store about animals and what possible actions are related to their welfare?
Henriette	Good question, we intend to keep various data about our animals, such as their name, age, species, relationship(s) to other animals, location in the zoo, diet, etc. There is more, but I don't know it by heart. Regarding the actions, we should be able to put new animals into the system and update their information when, for example, are moved to a different location, special diet, etc. Of course, we would like to eventually create timetables to manage their welfare. Think of their feeding timetable, when to move them their outdoor location, yearly health check-up, etc. To perform these tasks, we also need to assign our employees to do this. Ideally, this should be automated if possible but at least manually!
James	Okay so it is focused on addition and modification of animals, but I assume you also want to see the animals in various ways, for instance an overview of your animals, and in other occasions search for specific individuals?
Henriette	You are right, I forgot to mention this. We would also like to view individuals, but also aggregate information, or statistics about animals. How many animals do we have per species or location, view a list of animals indeed. We should be able to search for specific animals, as you mentioned, and look at their details and adjust their information.
James	You also mentioned that different types of timetables. To include all might be too much, which one has the highest priority and how would you do this manually?
Henriette	The feeding timetable would be of the highest priority. As for how, we would love to hear your ideas about what is best, as we now have multiple colleagues doing this on paper. They look at all the animals and their preferred feeding time and diet and then tries to assign one or more employees to do the task per zone. The problem we currently face is that is requires a lot of looking-up of information and error-prone. So, proposals are welcome!
James	I see; we are almost out of time, so I would like to conclude with one last question. It seems that various people will interact with the system. Is that correct and do you intend that there are differences between what they are allowed to do or see?
Henriette	Yes indeed, we envision different types of employees to use the system. The administration needs to manage our animals, the resource planners should only have access to the information they need to

	create the different schedules and our caretakers should be able to view data of the animals they are responsible. And as I mentioned before, there should also be a website for any type of employee.
James	Henriette, thanks a lot for your input, once my team is available, they will pick up from here.
Henriette	Great, I look forward to seeing the results of your work!

APPENDIX B: ASSESSMENT SHEET

Aspects	Poor/Unsatisfactory	Satisfactory	Good/Outstanding
Project plan	The questions from the reader are poorly answered. The group has not made a planning or the planning is clearly unrealistic.	The questions from the reader are answered to a satisfactory degree. The group understands what is asked of them and has a rough planning.	The questions from the reader are answered in an apt way, demonstrating full understanding of the problem space. The group understands what is asked of them and has a clear planning.
URS	The URS does not display clear understanding of the case. Insufficient elements.	The URS displays a sufficient understanding of the case. Use-cases are satisfactory, the minimal functional requirements are described.	The URS demonstrates how well the group understands the case. Use-cases handle most to all exceptions and the requirements are near complete.
Process report	The process report is (almost) empty. It is not following the guidelines from the reader.	The process report is partially updated during the project. However, it is incomplete.	The process report is clear about individual contribution and responsibilities. The individual reflections are self-critical. The report follows the guidelines from the reader.
Presentation	The presentation is unclear. There is no structure, the presenters are unclear to follow or the slides have "walls of text".	The presentation has a structure. The product is demonstrated where the main features have been shown.	The presentation has a good structure. The presenters have properly introduced themselves and the topics to be discussed. The demonstration feels professional and the presenters reflect both about what went good and bad.
Application	The software system has severe bugs that make them unusable. The M and S requirements have not been met.	The software system is fairly stable, there might be a few minor bugs. The M and S functional requirements have been implemented.	The software system is optimized, and has M, S and C functional requirements implemented. The system is robust and intuitive to use.
GIT	The group did not use Git or poorly.	The group used Git in a satisfactory way. They commited to the master branch and possibly experimented with branching. Commit messages clarify what changed in this commit.	The group uses Git in a professional way. They branch for feature development, standardize commit messages and use tags to identify intermediate stable versions. Everyone commits.
Client facing	The client is barely acknowledged, the interview is meager or non-existent. Major changes are not communicated towards the client.	The group interviewed the client with proper questions. Overal the client is informed about most major decisions, but might face surprises in the final presentation. The group does not hide all technical details from the client.	The group interviews the client in a concise way, asking questions that contribute to the better understanding of the client's requirements. The group communicates properly with the client informing and inquiring about important aspects, hiding (technical) details.
Tutor facing	The group has a passive attitude, communication with the tutor is kept short. The tutor is not made aware of progress. Internal struggles are hidden until escalated. Feedback is ignored.	The group has an active attitude, and communicates with the tutor. The majority of the progress is communicated, and most challenges are discussed with the tutor. Feedback is regularly taken oncorporated.	The group is pro-active. Communication is conducted in a clear format, meetings are pro-actively arranged and minutes are sent on time. Challenges are discussed with the tutor, and the group has investigated possible solutions. Feedback is reflected upon and incorporated.
Meeting conduct	There are no agendas, the group does not organize meetings in a timely fashion. Minutes are not sent after the meeting. The tutor has to actively manage the meetings.	The group organizes meetings at least a day ahead. Agendas are sent before the meeting and minutes before the next meeting. Agendas are satisfactory and minutes could be more to the point.	The group attempts to structurally organize the meetings. Where impossible, the group actively arranges meetings at least three workdays ahead of time. Meeting minutes are to the point and sent within a workday to the tutor/client.
Team work	The group does not work well together, giving rise to: miscommunication, not showing up for meetings, not meeting deadlines, etc.	The group collaborates sufficiently. There might be certain moments when progress stagnates. The group manages to stay on track.	The group collaborates well. Although one member organizes the group, the other members are also pro-active. Challenges faced are tackled by the group as a whole and reflected upon.

APPENDIX C: PROJECT PLAN AND USER REQUIREMENTS SPECIFICATION STRUCTURE

In the waterfall project you are expected to create two documents; a project plan and a user requirements specification (URS). The purpose of these documents is to clarify what you are going to do and how it will look/work. The project plan is intended to define the scope of the project and to plan ahead, whereas the URS (not an official document) is intended to design your software solution.

PROJECT PLAN

- **Client:** Who is your client?
  - How can you contact your client?
  - Which company does your client represent?
- **Team:** Who represents your group?
  - How can you contact this person?
  - Which team does this person represent?
- **Current situation:** What is currently known (contextually) about factors that influence your project?
  - E.g.: Are there competitors working on a similar solution? Is there an existing system that you build on? Etc.
- **Problem description:** What problem does the client have which they want to be solved?
  - It should not be the lack of solution.
  - If you know the solution, think about what problem your solution can solve?
- **Project goal:** What does your project attempt to achieve?
  - The problem describes lack of "something" (time, money, a clear overview, etc.) the goal describes the desired result.
  - A solution is not described here.
- **Deliverables:** What products are you going to deliver?
  - Documentation that supports the process are not deliverables (not to be confused with non-deliverables).
  - You only say what you deliver, don't describe your product(s) here.
- **Non-deliverables:** What are you not going to deliver?
  - There are a lot of things you will not deliver, but what might your client expect?
  - Were there requests by your client that you explicitly mentioned you will not deliver?
  - Might your client have expectations based on your deliverables list?
- **Constraints:** What are the project limits?
  - In order to keep the project realistic, you can define a scope for the project.
  - Limitations can be related to: time, money, quality, but are not restricted to these points.
  - What programming language are you going to use? How much money will you have to spend on possible hardware purchases? What are important deadlines? Etc.
- **Phasing:** How do you plan to conduct your project?
  - Phases: your project will have phases (timeframes) that conclude in project sub-deliverables.
  - Milestones: Phases conclude with milestones, deadlines where you finished sub-deliverables that can be reviewed by your client. These moments can result in go/no-go decision.
  - In the project plan you describe the planning towards your client (not your internal planning). So how many go/no-go decisions do you want?

USER REQUIREMENTS SPECIFICATIONS

- **Title page**
  - Title of document
  - Image
  - Date
  - Location
  - Members
  - Tutor
- **Table of content**
- **Agreements with client**
- **Functional requirements**
  - What should the system be able to do?
  - Group them by application/website (if applicable)
  - Prioritize them using MoSCoW (must, should, could, won't)
- **Use cases**
- **GUI**
  - Images of the various views.
  - A brief description per view how it works
- **Website wireframe (when applicable)**
  - The structure of your website with a brief description of its intent.

APPENDIX D: PROCESS REPORT

- **Title page**
  - Title
  - Image
  - Date
  - Location
  - Members
  - Tutor
- **Table of contents**
- **Work division**
- **Personal reflection**
  - Person A
  - Person B
  - Person C
- **Reflection about applying Waterfall**
  - What are its strengths and weaknesses?