

EduTrail

Vision of EduTrail

Revision History

Date	Version	Description	Author
06.03.22	1.0	First Iteration	Raphael Larsen, Nemanja Tosic, Sigrid Blystad, Daniel Sarjomaa, Sara Djordjevic
25.03.22	1.1	Second Iteration	Daniel Sarjomaa, Raphael Larsen, Sara Djordjevic
29.04.22	1.2	Final delivery	Raphael Larsen, Daniel Sarjomaa

1. Introduction

1.1 Purpose and scope

EduTrail is an application made with the purpose of aiding organizers in keeping track of posts, teams, points and times during rebus games, tasks which are typically carried out on paper or a spreadsheet document. It is the process of tracking points with EduTrail aims to streamline, while simultaneously making the games fun for both organizers and participants.

1.2 Definitions, Acronyms, and Abbreviations

Rebus: Rebus is a game consisting of multiple smaller games placed around a geographical area. Participants navigate using a map to find the games and complete them. The winner is determined by who has the highest total amount of points from all the games they visited during the run.

Checkpoint/Post: Two words used interchangeably, both referring to a geographical location on the game area with a minigame that makes up a small part of the complete rebus.

1.3 References

AltexSoft. (2020, December 1). Technical Documentation in Software Development: Types, Best Practices, and Tools. Retrieved from altexsoft:

<https://www.altexsoft.com/blog/business/technical-documentation-in-software-development-types-best-practices-and-tools/>

Baxter, K., Courage, C., & Caine, K. (2015). Understanding Your Users. Waltham: Elsevier Inc.

Bureau of Internet Accessibility. (2018, July 17). WCAG 2.1 Principles Explained: Perceivability. Retrieved from Bureau of Internet Accessibility:

<https://www.boia.org/blog/wcag-2.1-principles-explained-perceivability>

Capka, D. (n.d.). Lesson 4 - UML - Domain Model . Retrieved from ICTdemy:

<https://www.ictdemy.com/software-design/uml/uml-domain-model>

Christian, B. (2021, September 14). What is a Prototype Model in Software Engineering? Retrieved from Openxcell: <https://www.openxcell.com/blog/prototype-model-in-software-engineering/>

IBM. (2021, May 3). Messages in UML diagrams. Retrieved from IBM:

<https://www.ibm.com/docs/en/rsas/7.5.0?topic=diagrams-messages-in-uml>

Indeed. (2021, April 22). Use Case Diagram: Definition and Examples. Retrieved from Indeed: <https://www.indeed.com/career-advice/career-development/use-case-diagram>

Lucidchart. (n.d.). UML Sequence Diagram Tutorial. Retrieved from Lucidchart:

<https://www.lucidchart.com/pages/uml-sequence-diagram>

Mind Tools team. (n.d.). Mindtools. Retrieved from Risk analysis and risk management:

https://www.mindtools.com/pages/article/newTMC_07.htm

Nalimov, C. (2021, July 15). UML relationships explained: Dependency, Realization, Association, and more. Retrieved from Gleek: <https://www.gleek.io/blog/uml-relationships.html>

Rafique, I., Weng, J., Wang, Y., Abbasi, M. Q., & Lew, P. (2012). Software Learnability Evaluation. Beijing University of Aeronautics and Astronautics. Beijing. Retrieved from ThinkMind: https://www.thinkmind.org/articles/iccgi_2012_10_40_10358.pdf

Schroeter, E. (2021, November 23). What is Usability? (And How to Do It). Retrieved from CareerFoundry: <https://careerfoundry.com/en/blog/ux-design/what-is-usability/>

Segue Technologies. (2017, December 11). What is Usability and Why is it Important to Application Development? Retrieved from Segue Technologies: <https://www.seguetech.com/usability-application-development/>

smartdraw. (n.d.). Use Case Diagram. Retrieved from smartdraw: <https://www.smartdraw.com/use-case-diagram/>

Visual Paradigm. (n.d.). UML Class Diagram Tutorial. Retrieved from Visual Paradigm: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

Walker, A. (2022, March 22). UML Relationships Types: Association, Dependency, Generalization. Retrieved from Guru99: <https://www.guru99.com/uml-relationships-with-example.html>

West Virginia University. (2000). Chapter 8: Sequence Diagram. In UML Tutorial for C++ - Windows Platform. West Virginia: Advanced Software Technologies, Inc. Retrieved from <https://community.wvu.edu/~hhammar/rts/Chapter8-Sequence%20Diagram.pdf>

S.A Amirshahi

1.4 Overview

The rest of the Vision Document contains an overview of how the final product will be, what kind of features and functionality it will have. It will also describe who the product is tailored towards, it will contain a risk analysis and estimated costs for the project.

2. Positioning

2.1 Business Opportunity

Our product fills a niche role, and therefore faces little to no competition. This lightweight application is specifically tailored to run the rebus game and would be convenient for organizers that administrate these games on a regular basis. The simple nature of the program also means that the cost of maintenance will be close to naught, resulting in a fairly low risk project.

2.2 Problem Statement

The problem of	managing the continuous flow of results during a game of rebus
affects	the administrators of the games
The impact of which is	results may not be registered in a correct and timely manner, and the process may be stressful for administrators.
A successful solution would be	to streamline the point counting process, making it easier for an administrator to quickly add points during the game.

2.3 Product Position Statement

For	rebus administrator
Who	dislike the hectic nature of being a rebus administrator.
The EduTrail application	is a point, team and game management tool
That	offers a more streamlined approach for managing a large game of rebus.
Unlike	the pen and paper or excel alternative
Our product	is tailor made for rebus.

3 Project goals

3.1 Impact goals

Our aim for this project is to make the Rebus game more efficient, streamlined, and simple for organizers, thus making the game more enjoyable for all parties involved. The specific goals we hope to achieve with our application are as follows:

Facilitate hosting of larger events

Increase efficiency

Reduce result errors

Increase convenience

Reduce stress

When administering a rebus game of any notable size, there is often a significant amount of data being rapidly sent in from individual game admins to a central scorekeeper admin. Our product aims to make the scorekeeper role less stressful by increasing the efficiency of writing down results. Instead of fumbling through a spreadsheet document to find the correct checkpoint and team, the admin can now simply input the team number, checkpoint number, points won, and press enter.

The improved efficiency and ‘railroading’ nature of the application also eliminates the possibility of writing results in the wrong spreadsheet cell, thus making the data of the overall game more reliable. The previously mentioned goals also include the overarching goal of improving convenience. A modular save system, spreadsheet exporting, and rapid point registration makes for a more convenient solution to scorekeeping in rebus. Finally, the improved efficiency and convenience our application provides will open the possibility for even larger rebus events without increasing the number of scorekeepers.

3.2 Result goals

The goal of this project is to develop an executable program for computers with supporting features to achieve the goals listed above. The specific features we want to include in the final product:

Terminal User Interface

Add, edit and delete checkpoints

Add, edit and delete teams

System for rapid input of results

Saving and loading of teams and checkpoints

Export scoreboard as CSV spreadsheet

You can read more about the features of the application in the ‘Features’ section.

The final deadline of the project is set to April 29th, 2022. The final product will include the executable program as well as a user manual and documentation. Source code may also be

included depending on the client's request. Leading up to the final release, there will be at the least two prototypes which will be presented in future client meetings.

As for resources, we aim to have the project completed within about 550 total hours of work.

If the product is finished ahead of schedule, we hope to expand the application with more features that will be decided in a client meeting.

3.3 Process goals

Apart from the product itself, there are several goals we want to achieve, most of which boil down to getting a good grade and developing our skillset for future projects. We specifically wish to get a good grasp on the process of developing an application in the most efficient and painless way possible, while still creating a product that lives up to the customers' expectations.

4. Stakeholder and User Descriptions

4.1 Market Demographics

EduTrail does not necessarily have a specific target demographic considering the nature of the sport. Administering a Rebus game is as popular with young people as with the elderly and there is not any need to choose a key age demographic as this would just reduce maximum market potential. Geographically speaking rebus racing is far more popular in Europe and North America compared to other continents, and therefore EduTrail will primarily be developed with western markets in mind. Calculating total market size for such a niche market is not possible, but with little to no competition EduTrail could have a significant impact on market growth.

4.2 Stakeholder Summary

Name	Client
Description	Contacted our dev team
Type	Advisor
Responsibilities	Guide the dev team Give constructive feedback throughout the project

	Review the final product
Stakeholder's criteria	Finish product before deadline Accomplish features requested by the client Receive a bug free and usable program according to the wireframe Do not go overbudget

Name	Development Team
Description	The team working on EduTrail
Type	Programmers
Responsibilities	Listen to feedback from users and client and then change the program according to their needs Ensure that the program is bug free and working Make sure that the source code is understandable for any future programmer working on it
Stakeholder's criteria	Receive constructive feedback from other team members, test users and client

4.3 User Summary

4.3.1 Test users

Name	Test users
Description	Users that volunteer to give constructive feedback by testing
Responsibilities	Help the development team by testing our products and come with helpful information we could use to enhance our product
Stakeholder	The dev team is responsible for representing this user type's interest

4.3.2 End users

Name	End users
Description	Users that use our program and that potentially can give constructive feedback
Responsibilities	Use our program in a non-harmful way
Stakeholder	The test users are used to represent an end user's interest

4.4 User Environment

EduTrail is intended to be used on a single device at a time. Our program would allow for multiple people to use the program on the same device, but it will not be possible to administrate the same race between multiple devices. The task cycle for EduTrail will highly depend on how long the rebus race lasts and will most likely take 5 minutes to 1 hour to complete depending on the size of the rebus tournament. Since EduTrail will use a Terminal console interface it will be limited to PC and laptop use only and this can make it challenging to use in a lot of places. Weather conditions such as rain, snow and direct sunlight will make it difficult to administrate a rebus race outdoors. The main platform for EduTrail will be Windows 10 but it will most likely work with older versions such as Windows 7 and 8, and the newer version Windows 11.

4.5 Key Stakeholder or User Needs

Need	Priority	Concerns	Current Solution	Proposed Solutions	Completed
Multiple point types for a post	Medium	Would require a lot of work and the whole point system would have to be reworked	Single point-scoring type. A user set number between 1 and 99	No solution proposed, therefore current solution will be the final solution	no
Output is too hard to read	Medium	No concerns	Text is displayed line after line with	Clear out terminal screen after each menu option is chosen so that	yes

			little to none spacing	there is only one menu at a time visible to the user.	
--	--	--	---------------------------	---	--

4.6 Alternatives and Competition

1. - Microsoft Excel

Excel spreadsheets could be perceived as a competitor for our product. Excel's major strengths are a degree of versatility that caters to a wide range of problems, a variety of hotkeys which allow for efficient input and functions that reduce the user's workload. Multiple user access is also a great strength. The weakness of Excel is that it's not streamlined for setting up a rebus race, therefore some knowledge in excel is required to be able to make a working race. Since Excel is a general-purpose application, its variety of features may be more redundant than they are useful. For those who are not particularly tech savvy, Excel may be too advanced, and its many features may have a discouraging effect.

5. Product Overview

5.1 Product Perspective

While EduTrail is a standalone application, we expect the user to utilize spreadsheets and other traditional methods for keeping track of points in parallel with our program. It is therefore important for us to have files in a format so that data might easily be transported between applications. For this we have chosen CSV for its simplicity and compatibility. Utilizing a standardized format like CSV will make EduTrail integrate seamlessly into the environment it occupies.

5.2 Summary of Capabilities

Customer Benefit	Supporting Features
Administrators can quickly allocate points to teams.	Simple GUI results in efficient navigation.

Easy installation.	The executable is completely standalone and requires nothing more than a folder to store saved files.
Easy extraction of result and compatibility with spreadsheet programs.	The program can generate a CSV spreadsheet with results from a game.

5.3 Assumptions and Dependencies

Although it is expected that the simplicity of EduTrail will make it standalone, we cannot exclude possible unforeseen dependencies that may alter the applications functionality.

Possible dependencies include:

Windows 10 OS. As of now, we have not planned support for any other OS than Windows 10. While this does not necessarily mean that EduTrail will not work on other operating systems, we will not dedicate resources to do the necessary testing required to make EduTrail compatible with alternative OS.

5.4 Risk analysis

5.4.1 Risk Assessment					
ID	Risks	Risk description	Consequence	Likelihood	Risk Rating
1	Team member becomes sick/injured	The likelihood of a team member becoming sick or injured is almost certain and would lead to minor consequences. There is not much that can be done to reduce this risk. But by taking precautions at meetings could hinder any of the other members to also get sick	Minor	Almost certain	High

2	Loss of motivation	Loss of motivation could lead to a much slower workflow in the group and could go on for a while therefore the consequence of this is moderate and the chance of this happening is likely.	Moderate	Likely	High
3	Technical issues with computer	A computer can break down in multiple ways and would often lead to it being unusable. A computer not working won't necessarily have that big of a consequence since we backup our files and it's quite easy to use a reserve computer in the meanwhile. Therefore, the consequence is only moderate with a likelihood of possible.	Moderate	Possible	Medium
4	Loss of data	Loss of data can happen if there's no backup and someone accidentally deletes, or the hard drive goes bad. The likelihood of this happening is possible, and it would lead to major consequences. The risk of this happening could be reduced by making sure everyone saves new	Major	Possible	High
5	Run out of funding	Running out of funding would have severe consequences since it would immediately halt our project and the likelihood of this happening is unlikely. It's unlikely to happen because we track our hours and always divide up tasks effectively as to be as effective as possible with our time budget.	Severe	Unlikely	High

6	Team member dropping out of the course	The risk of a team member leaving the project is unlikely since our team are all very motivated students, but if it were to happen it would lead to severe consequences. There is no way to remedy this because either way a team member leaving will remove 20% of our workforce and would also lead to other repercussions that would slow down our project even more.	Severe	Possible	Extreme
---	--	--	--------	----------	---------

5.4.2 Consequence

Descriptor	Level	Definition
Insignificant	1	Will not lead to any delays or setbacks
Minor	2	Can lead to a small setback
Moderate	3	Project will slow down until a solution is found and can lead to a moderate setback.
Major	4	Project can be put on hold until solution is found, and will lead to a major setback
Severe	5	Project will be put on hold for person experiencing the problem until solution is found, and a severe setback will occur

5.4.3 Likelihood

Descriptor	Level	Definition
Rare	1	May occur sometime (“once in a lifetime / once in a hundred years”)
Unlikely	2	May occur sometimes over an extended period of time
Possible	3	May occur several times over a period of time
Likely	4	May be anticipated multiple times over a period of time

Almost 5 Prone to occur regularly
Certain

5.4.4 Risk Level/Rating and Actions

Descriptor	Definition
Extreme:	Notify Client and Team leader immediately. Corrective actions should be taken immediately.
High:	Notify Team Leader immediately. Corrective actions should be taken within 24 hours of notification.
Medium:	Notify Team Leader. Follow up that corrective action is taken within 3 days.
Low	Notify Team Leader. Make sure that corrective action is taken within a reasonable time.

5.4.5 Risk Matrix – Displays where each risk id is located on a risk matrix.

Likelihood	Consequence				
	Insignificant	Minor	Moderate	Major	Severe
Almost Certain		ID.1			
Likely			ID.2		
Possible			ID.3	ID.4	ID.6
Unlikely					ID.5
Rare					

(Mind Tools team, u.d.)

5.5 Cost, pricing and benefits

The benefit of the project is that the application will be delivered within the specifications that the client requested. The client needs EduTrail for a very specific task, and apart from

succeeding in that task, there is no additional benefits. The profit margin of this project is practically zero, which of course is expected, as the hourly rate of development is only to cover developer salary and miscellaneous costs.

5.6 Estimated Costs

As have been stated previously, this project has a low cost of maintenance. A breakdown of the project costs can be found below. The cost of development is considered a 1599kr hourly rate per hour. 1599kr multiplied by the 550 designated work hours results in a total price of 879 450kr.

Feature	One-time Expense	Continuous Expense
Development	879 450kr	<1000kr per year
Download Hosting		<1000kr per year

All currency is in Norwegian Kroner (NOK).

6. Product Features

EduTrail is an application that is made according to requirements of the client and allows easy, precise and clear management of a Rebus game. Based on researching the needs of the program for these purposes as well as the functions required to make the registration of the games like this easy and simple, we have provided the following application features:

6.1 Terminal User Interface / Main menu

The program is based on standard text-based user interfaces (TUI) which allows easy and precise data entry, as well as their review. After launching the application, the user accesses the main menu with options. Different option further defines the method of entering the desired data. This allows user quick and simple navigation through application.

6.2 Managing checkpoints

By selecting this option from the main menu, the user accesses the data related to the checkpoints in the game. These checkpoints are places that teams/individuals must visit during the game. Based on the time for which the places were visited, the teams/individuals get points. Through this option, the user can enter new checkpoints, modify or delete existing checkpoints and thus create a game of his choice.

6.3 Managing teams

This option allows the user of the program to register future competitors, whether they are teams or individuals. It is also possible to enter data of individual members of a particular team. Access to further sub-options such as adding, modifying or deleting existing teams (and their members), or individual competitors.

6.4 Managing points

As in any game, awarding points and scoring teams/individual participants is very important so that the teams/individuals at the end of the duel can be compared with each other. We also need an insight into which team/individual has won. This option allows user to easily register points for each team or each individual participating in the duel. Based on the data entry for passing through the given checkpoint and the time for which it was done, the teams get points.

6.5 Showing Results

Option that allows review of points scored by teams/individuals based on time spent for a particular checkpoint. The user can quick and easy see the number of points scored for a certain period of time for each team/individual in the duel. Also, option provides insight in total points scored on(for) a particular checkpoint as well as the insight of the total number of points for each team/individual for current duel.

6.6 Settings

Here the user can create a new duel/tournament according to the desired criteria. Also, the user can save data for each duel/tournament played. This enables later access to the data of all

duels played and their use if it is necessary in the future (comparison of results, success of different teams/individuals by points, difficulty of given checkpoints, team statistics etc.).

7 Constraints

In this project the main constraints we must take is all linked to our somewhat lacking skillset. Most of the team have completed a basic programming course and is currently learning object-oriented programming in C++, but that is as far as our abilities go. We have little knowledge in databases, networking, GUI implementation and other tools that is often instrumental in modern applications. What we are left with is a close to stock version of C++. The project will however not suffer from these constraints, as we simply plan and design the application around our skillset. For example, that is why the program will have a terminal/command line-based user interface.

8 Quality Ranges

Edutrail will be programmed to be robust, meaning that there should be no way that the user can write a wrong type of input such as a string in an int input and ensures that it doesn't lead to crashing the program. Edutrail will be fault tolerant and will make sure that any possible error will be caught by an "if check" before the error occurs. Edutrail will have good usability by creating a simple, yet easy to understand GUI with effective control of the menus.

9 Precedence and Priority

EduTrail in its final version is supposed to have several features, which in a worst-case scenario may not all be possible to complete. Below is a ranking of what features takes highest priority.

1. Adding checkpoints and teams. System for point registration.
2. Editing of teams and checkpoints.
3. Rapid score input system.
4. Time based checkpoint types.
5. Resetting points.
6. Basic saving/file writing.
7. Export results as spreadsheet.

8. Modular saving of checkpoints and teams.
9. Hybrid checkpoint types. (Not required)
10. Automatic SMS point registration (Not required)

The features listed at the bottom of this list are not required by the client but is something that is requested if we finish the base product far ahead of schedule. We do not expect to implement these features.

Otherwise, the basic systems for adding teams, checkpoints and giving them scores have the highest priority. Further down are the systems for further customization of checkpoints and methods of saving, loading and exporting data from the application.

10. Other Product Requirements

10.1 System Requirements

To run the EduTrail application a Windows 7 or newer operating system is required to run the .exe program. The hardware requirements to run the application file is lower than the requirements to run Windows. Therefore, to run Windows and EduTrail the bare minimum requirements are as follows:

- 1 gigahertz (GHz) or faster 32/64-bit processor
- 1 gigabyte (GB) RAM (32-bit) or 2 GB ram (64-bit)
- 16 GB available disk space for the OS + ~5 MB for EduTrail
- DirectX 9 or newer supporting graphics device

The required peripherals to run and use EduTrail is a keyboard and mouse as well as a monitor

10.2 Performance Requirements

EduTrail should run reliably with few performance issues related to throughput, user load, accuracy and response times.

Expected performance requirements:

- The application should never crash or quit unexpectedly
- Precise and accurate performance (Output should not change if the same input is given multiple times)
- Response time under 1 second.

10.3 Environmental Requirements

For software applications such as EduTrail, environmental factors include user environment, resource availability and error handling. By default, a user in a user environment has access to opening most type of files and applications. But, in very rare circumstances permission to running .exe files could be disabled by a network administrator. EduTrail's resource requirements are second to none, therefore the program would only stop running if 100% of the system's resources are unavailable. Any errors occurring while running EduTrail could happen because of corrupt registry settings or system issues due to virus infections. Since there are zero expected errors from the program itself the only way, errors could happen if there is a problem with the host computer. Therefore, the user should make sure registry settings are correct and make sure no virus is infecting their pc.

11 Documentation Requirements

11.1 User manual

The user manual's main purpose is to give assistance to the user on how to use the EduTrail application but should also be creating a better user experience. The manual should contain:

- Product name
- Index
- Description of user interface
- Installation instructions
- How to use the product
- Troubleshooting section in case of error
- Glossary
- Contact details in case of further support

The length of the manual shouldn't be too long to overcomplicate it but should be about 1-2 pages long depending on text size and format. The EduTrail application is simple, Therefore, a long user manual isn't necessary. The manual should be precise and simple since the application already is easy to understand and an advanced manual would be harder to understand than just running the application alone. Which effectively diminishes the purpose of a manual. An index should also be added so the user easily can find the right page and a glossary with special, unusual, or technical words or expressions in an alphabetical list with an explanation for each word. A full user manual is more useful than just a tutorial because the tutorial only helps with using the product and doesn't account for any errors or other problems with the host computer. Whereas a user manual will help with troubleshooting in case of errors and download instructions to minimize the risk of faulty installation.

11.2 Read Me File

The Read Me File for our application will contain a "What's new with this release" section, and it will discuss compatibility issues and define known bugs if any exist in current or earlier versions.

11.3 Labelling and Packaging

When running the EduTrail application the logo in ASCII text will be displayed once

12 Feature Attributes

12.1 Status

Proposed	SMS Input Modular Saving/Loading Hybrid Checkpoint Types
Approved	Resetting All Points Viewing Team Details Viewing Checkpoint Details CSV Exporting

Incorporated	Adding teams, checkpoints Basic Point registration system Quick point registration system Displaying Scoreboard Saving/Loading
--------------	--

12.2 Effort

The features previously listed are the result of a discussion we did at the beginning of the project of what was possible to achieve within the time frame. Obviously, the project has fallen behind schedule, and it is clear a better risk analysis could have helped us averse the challenges that put us in this position.

At the early stages of the project the intention was to have 2-3 people working on coding the product. However, after a few weeks of development it became clear that the amount of work needed elsewhere in the project exceeded our expectations, and as such we eventually cut down to a single person working actively on the code.

At this point we were already behind schedule, and a single developer was able to work more efficiently to complete a viable product while other team members were able to put more attention on user testing and documentation.

12.3 Target Release

The release is set for 29th of April 2022, and there is no flexibility in this date. As such, the application will be released whether it is ready or not. During the last client meeting it became clear that we had fallen behind schedule and was asked to cut some features. While lacking some features, version 2.1.7 has currently no known bugs and will likely be the release version.

As noted in the status section, there is multiple approved features that have yet to be implemented in the product. With little time remaining until the release, it is unlikely that further resources will be put into implementation of new features.