Project Report

CoPilot application

Authors:
Tommy Ernsund
Viking Forsman
Joaquín García Benítez
Iván Muñiz
Mathias Svensson Karlsson
Clara Torre García-Barredo

Introduction

This project report is part of the course *DVA313 - Software Engineering 2: Project Teamwork* at Mälardalens University. During this course students get to develop a software product for an external client.

Volvo Construction Equipment

The client in this project is Volvo Construction Equipment (Volvo CE), a major international company that develops and manufactures construction equipment and vehicles. They are currently developing autonomous vehicles that can be programmed to follow predefined paths and to perform specific tasks.

The CoPilot app

The client has asked for an Android application that should work both on mobile phones and on the CoPilot, which is an integrated tablet in the cabin of a vehicle, running a Volvo internal version of Android. This tablet's main purpose is to control the autonomous machines they are developing. The application they have requested should provide the functionality to plan the paths and actions of the autonomous vehicles on a construction site, or any other workplace they decide to use them on. A path consists of a list of sequential points. Each of these points represent actions that will be performed by the machine.

The user adds points to the path using their mobile device. These points can be added using the user's current position as input, or by placing a new point manually in the map view. Once a path is completed it can be executed using the CoPilot app. The paths can be seen both with a visual representation and in G-Code. G-Code is an archaic CNC programming language that is often associated with computer aided manufacturing.

Contents

Organization

The development team

The development team consists of six developers.

- Tommy Ernsund, bachelor student of Computer Science at MDH University.
- Viking Forsman, bachelor student of Computer Science at MDH University.
- Joaquín García Benítez, bachelor student of Computer Engineering with focus on Computation at Universidad de Cantabria with an exchange program in MDH University.
- Iván Muñiz, bachelor student of Computer Science at Instituto Tecnológico y de Estudios Superiores de Monterrey with an exchange program in MDH University.
- Mathias Svensson Karlsson, bachelor student of Computer Science at MDH University.
- Clara Torre García-Barredo, bachelor student of Computer Engineering with focus on Computation at Universidad de Cantabria with an exchange program in MDH University.

All members of the development team was required to contribute in the three main parts of the project: the implementation, the documentation and the presentation of the product. Tommy Ernsund, Mathias Svensson Karlsson, and Iván Muñiz had previous experience with Android Studio, and served an advisory role on matters regarding the usage of this software. Joaquín García Benítez was responsible for leading the presentations during the meetings with the steering group, while Viking Forsman and Iván Muñiz were responsible for taking notes on any feedback. Clara Torre García-Barredo was responsible for managing all communication with the client and the project Boards, with the exception of client meetings, where every team member was actively involved.

Team routines and synchronization

The project version control was handled with the web-based hosting service GitHub. This same service also provided a Kanban board for project planning. This provides every developer with the opportunity to see the project backlog at any time, and choose which task to focus on next. Internal communication was provided by the communication service Slack, in which the conversations are divided into different channels for easy access.

There were weekly meetings scheduled with the client, during which the developers presented product increments to get feedback on. These meetings were held over the communication app Skype for Business, since it required less time than gathering the development team and client in the same location. Volvo CE's headquarters are located in Eskilstuna while the university campus is located in Västerås, and the traveling time between these locations is about one hour by bus. Skype for Business also has a screen sharing functionality that allowed the team to easily present any updates that had been made for the CoPilot app during that specific week.

There was also weekly meetings with the steering group, where any issues or obstacles encountered by the team were presented. Before any of these events, the entire development team collectively prepared the presentation, questions and topics to cover during any of the meetings.

Deviations from the project plan

There were slight deviations from the project plan due to the initial work estimation being too optimistic. The deadlines for the deliverables had been set three days before the officially assigned course due dates, and that was not always successfully accomplished.

During the two final weeks of development it was hard to find a time and location for the meetings that suited everyone in the development team. The meetings then moved on to be via the communication service Discord, instead of being in person at the campus in Västerås. This method lead to some misunderstandings, but it was deemed preferable than having no meetings at all.

Development Process

Desired product features

What's being developed is a prototype of the CoPilot app, which Volvo CE's team will develop further once the initial implementation is done. The developed product has only implemented the functionality related to the viewing, creation and edition of paths. The finalized product will be integrated into Volvo CE's internally built Android SDK. It will also feature extended functionality, such as being able to connect with the autonomous machines and instruct them to follow a predefined path. For a more detailed description, see the Project Plan and Detailed Design Description documents.

Project deliverables

Document	Deadline	Submitted	Description
Project Plan	2018-11-22	2018-11-22	Document that defines the project development plan, including diagrams and a Gantt chart. It also contains Use cases that defines the project requirements.
Detailed Design Description	2019-01-17	2019-01-17	Document that defines the projects architecture, important design decisions and external system and frameworks used in the project.
Product	2019-01-17	2019-01-17	Implementation of the product which was defined in the project plan and detailed design document.
Project Report	2019-01-17	2019-01-17	Document that describes the project outcome, both in terms of results produced and experiences.

Table 1: Deliverables' submission dates

Reported work hours

Developer	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	Total
Tommy Ernsund	17.5	11.0	11.8	15.2	21.3	5.5	0.0	12.7	22.8	7.9	125.7
Viking Forsman	17.5	11.3	19.2	12.0	14.2	6.8	0.0	21.7	26.4	5.0	134.1
Joaquín García Benítez	20.7	13.5	15.8	12.7	8.2	13.8	5.5	15.2	20.5	17.8	143.7
Iván Muñiz	17.5	8.0	10.2	15.7	6.7	2.0	0.0	0.0	20.2	7.6	87.9
Mathias Svensson	18.5	12.0	24.0	15.4	11.8	10.0	6.3	19.2	38.4	12.0	167.6
Clara Torre García-Barredo	20.7	13.5	19.5	13.3	7.7	12.7	1.3	14.3	19.0	15.8	137.8
Total	112.4	69.3	100.5	84.3	69.9	50.8	13.1	83.1	129.1	64.8	796.8

Table 2: Hours spent developing the project (rounded to the closest decimal)

Work distribution on major project features

This is a table of features with the main developers associated. It's important to note that most developers have worked on all features to a lesser degree, and that this list doesn't cover bug fixing.

To the bug fixing extent, Mathias Svensson Karlsson and Joaquín García Benítez were the developers most involved in this particular task.

Major product feature	Associated developers
User authentication (Login and logout)	Clara Torre García-Barredo, Joaquín García Benítez
Path list view	Mathias Svensson Karlsson
Map view	Mathias Svensson Karlsson, Tommy Ernsund
Creation and editing of path	Mathias Svensson Karlsson, Tommy Ernsund
List filtering	Viking Forsman
Machine view	Iván Muñiz, Viking Forsman
GCode view	Iván Muñiz, Viking Forsman
Firebase database	Iván Muñiz, Mathias Svensson Karlsson
Firebase cloud functions	Mathias Svensson Karlsson
Toolbar menu	Clara Torre García-Barredo, Joaquín García Benítez

Table 3: Major product features and the developers involved. For a more comprehensive view on involvement, the commitment history of the project's Github repository is suggested.

Testing

System testing

The system testing has been developed, run and passed when a feature is created and integrated with the app. The system testing should pass different requirements, chosen by the team. Also, if one of the tests is not passed, the feature is revised until the problem is solved.

Acceptance testing

The acceptance testing was carried out at Volvo CE's headquarters in Eskilstuna, with the client and entire development team present. Specific instructions on how to perform the test cases used in the acceptance testing can be found in appendix 1.

During the acceptance testing we tested the following functionalities:

- Registration and login of user
- Search after path by name or description
- Delete paths
- Create paths
- Edit path name and/or description
- Edit paths
- Assign path as favorite
- Display multiple paths simultaneously
- Sign out user

Acceptance testing outcome

The acceptance testing took place at the Volvo CE office in Eskilstuna. It was attended by three Volvo CE workers, including the client contact, and the development team. The representatives from Volvo CE deemed that the CoPilot app met the product requirements, and they also gave suggestions for further improvements.

Some of these suggestions have been implemented in time for the product deadline, like the "Points of Interest". This feature consists of a label that can be applied to any point in the path, and marks it as a point to be referenced in the future. It has been thought of as a reference for the planners rather than for the machines.

The rest of the suggestions are unfortunately not implemented due to lack of time.

Result

Missing desired features

A possible improvement to the current app is to add decision points to paths. This feature was suggested by the client in the later stages of the apps development. But the development team deemed that this feature wouldn't be able to be implemented within the deadline of the project, since it would require extensive rework of the product base classes.

Suggestions for improvement

There are also some suggested improvements for further development of the CoPilot app.

GUI

Some parts of the app's user interface could be improved. For example, the tablet version could be developed further by adapting it to landscape mode and making better use of larger screens.

Functionality

- There's a need to handle service failures, such as no internet connection. At the current time there is no implementation taking this into consideration.

- Currently all allocatable machines are drawn from the same pool. This might not be the optimal solution, since a company might own a multitude of autonomous machines who are stationed in different quarries or construction sites. Therefore it would be an improvement for operators to first choose which quarries or construction site to allocate machines from before creating a path.

- Possibility to hide markers in the map view.

GPS

The finalized version of this app would also require a more accurate way of measuring the user's position, because the mobile GPS can have an error threshold of multiple meters. This threshold would not suffice in a real world setting, where the autonomous machine following the path could get damaged or cause harm to other machines or personel.

An alternative to regular GPS is the technology "real time kinematic" (RTK) which works by placing stationary GPS devices at fixed GPS-coordinates, that error-correct any coordinate offset. Furthermore there is "Network-RTK" which works on the same principle, but uses more stationary GPS devices to create a virtual space within the designated area. In Sweden in particular, you have access to the nation wide network-RTK known as SWEPOS for free. There are however drawbacks with RTK, such as being ill suited for rough

terrain. A more global alternative to RTK would be Precise Point Positioning (PPP), albeit a little less accurate, with an accuracy of 3cm. PPP is a paid service, and works on the same principle as RTK, but instead of having local GPS devices, the corrections are already made by PPP itself by communicating with various ground-stations around the globe. This service might be interesting to Volvo CE, being a global company.

Conclusion

Positive experiences

The team has encountered a lot of learning experiences during the implementation of this project. We have gained experience on how to work in larger groups and how to organize work effort towards a common goal. We have learned a lot of practical skills in regards to utilizing modern development tools such as GitHub and Slack more effectively. It was also an exciting opportunity to work for a real client from a large international company such as Volvo CE.

Things that could been handled differently

There are some things that, in retrospect, could have been handled better. In the initial phases of the project, there was a problem regarding time reporting; this problem persisted for the first two weeks of the course. Initially the time reporting was measured in minutes rather than hours which made it hard to survey. Furthermore, column charts were used to present these during the steering meeting on a per day basis, which made readability even worse.

Advice for future students

Lastly there are some concrete advice for future students taking this course:

- Try to be realistic when estimating the required work effort for project tasks. Keep in mind that only half of the disposable time can be directed towards the project, since the course is given at half-time.

- Time is a limited resource in this course, so it’s important to focus on the part that will make your project functional. Only start adding extended features after implementing the base functionalities, given that enough development time remains.

- Try to have weekly meetings with the client, so that feedback can be received in the early stages of development. This will help to avoid situations where you have to do major rework to fulfill the client’s expectation of the project.

- Practice with other team members before presenting the project deliverable. It can be hard to perform the presentation within a given timespan, so it is good to come prepared.

Appendix 1: Acceptance test cases

This is a detailed description of the test cases that were used in the acceptance testing of the CoPilot app:

ID:	Test case 1
Title:	Registration and login of user
Description:	User can register a new account, which can then be used to login to the app.
Pre-conditions:	1. User is not logged in 2. Unused email address is used when registering the account.
Post-conditions:	1. A new user is registered.
Scenario steps:	1. User enters email address. 2. User enters password. 3. User clicks the *create account* button in the start screen. 4. System adds the new user to the database. 5. User clicks the button labeled *take me back to login*. 6. User clicks login.

ID:	Test case 2
Title:	Search after path by name or description
Description:	User can search after a specific path by name or description in the pathlist view.
Pre-conditions:	1. User is logged in
Post-conditions:	1. Only paths that contains the search query are displayed in the path list.
Scenario steps:	1. User clicks search icon in the toolbar. 2. System expands search field. 3. User enters query in the search field. 4. System filters the path list so that it only shows paths that contain the search query.

ID:	Test case 3
Title:	Delete path
Description:	User can delete specified path
Pre-conditions:	1. User is logged in. 2. User has administrator status. 3. There is at least one item in the path list.
Post-conditions:	1. The selected path is permanently removed from the database and local device.
Scenario steps:	1. User selects the path they wish to delete. 2. User clicks the the *delete* button. 3. System shows confirmation dialog. 4. User clicks the ok button. 5. System removes path from the database and local device.

ID:	Test case 4
Title:	Create path
Description:	User can create new paths.
Pre-conditions:	1. User is logged in.
Post-conditions:	1. A new path is added to the database and local device.
Scenario steps:	1. User clicks the *create* button. 2. System displays the map view. 3. User presses the *start point* button. 4. System checks user's position and adds start point to path. 5. User clicks any of the *point type* buttons. 6. System adds point to the path if it is valid. 7. Repeat step 5 and 6 an arbitrary number of times. 8. User presses the *end point* button. 9. System adds point to the path and adds the path to the database.

ID:	Test case 5
Title:	Edit path name and/or description
Description:	User can edit the specified paths name and/or description
Pre-conditions:	1. User is logged in. 2. There is at least one item in the path list.
Post-conditions:	1. The selected path's name and/or description is edited in both the database and in the local device.
Scenario steps:	1. User selects the path they wish to edit. 2. User clicks the the *edit* button. 3. System shows edit dialog. 4. User updates the path name and/or description. 5. System changes the path information and updates the database.

ID:	Test case 6
Title:	Edit path
Description:	User can edit the points in an existing path.
Pre-conditions:	1. User is logged in. 2. There is at least one item in the pathlist.
Post-conditions:	1. The points in the selected path are edited in both the database and in the local device.
Scenario steps:	1. User selects the path they wish to edit. 2. User clicks the edit button. 3. User chooses the changes that they want to perform. 4. New changes are saved to the database and in the local device.

ID:	Test case 7
Title:	Assign path as favorite
Description:	User can assign a path as a favorite. Favourite paths will be placed at the top of the pathlist.
Pre-conditions:	1. User is logged in. 2. There are at least two items in the pathlist that are not assigned as favourites.
Post-conditions:	1. The selected path is assigned as a favorited, and will be placed above regular paths.
Scenario steps:	1. User selects the path they wish to assign as a favorite. 2. User clicks the *favorite* button. 3. System updates the favorite icon, which is displayed in the expanded listview. 4. System changes the order of items so that the newly favorited path is above any item that isn't favorited.

ID:	Test case 8
Title:	Display multiple paths simultaneously
Description:	User can view multiple paths simultaneously in the mapview
Pre-conditions:	1. User is logged in. 2. There are at least two items in the pathlist
Post-conditions:	1. Display multiple paths in the map view
Scenario steps:	1. User selects the first path they want to display. 2. User selects edit path. 3. User swipes left from the right side of the screen. 4. User selects second path to display. 5. System changes in order to display both paths.

ID:	Test case 9
Title:	Sign out user
Description:	User can sign out from the application, preventing any unauthorized personnel from using the application's functionality.
Pre-conditions:	3. User is logged in.
Post-conditions:	2. User is logged out.
Scenario steps:	6. User clicks the the menu button on the toolbar. 7. System displays drop list of options. 8. User clicks the option labeled *sign out*. 9. System signs out user. 10. System changes view to the login screen.