|  |  |
| --- | --- |
| **Ersan** | **Presentation startet**  Good morning everyone, This is Group A and Jan Leonardi, Karsten Rudolf, Edgar Meilinger, Armine Amzil and Me, Ersan Ünsal are here today to show our Project for Advanced Programming. Our examiner is Jürgen Jung.  **Nächste Folie**  In this Presentation we will give a short overview of our Project, starting with the provided main Tasks from our supervisor. Following by the Requirements and our Architecture, we build to fulfil the given Tasks.  Also, we will show you how we worked during our Sprint planning and at the end of the presentation we will provide you an overview of our main Features in our Application. And then we are very happy to show you the live Version and guide you through every Use Case we prepared. |
| **Karsten** | **Task**   * Lets start with our first point of the agenda: the task * The task was to develop an application landscape tool * With the following User Stories * Applications can be created and displayed on the canvas * Each application has it’s own properties * In our case: Name, Description, COTS, release- and shutdown date * Also being able to create a dataflow between applications * You can add multiple data objects to those data flows * So each data object also has it’s own properties: Name, description, personal data   **And the whole diagram, with all its applications, data flows & data objects and their respective properties can be saved persistently into a database** |
| **Jan** | On the next slide you can see our software architecture. GoJS was requested by the customer, so that was included as a requirement. Our program runs in the open source JavaScript runtime environment NodeJS. For the frontend we use the framework ElectronJS which was written with PUG, CSS and Bootstrap. The frontend communicates with the backend via Rest-API. The backend includes the ExpressJS server-side web framework and processes XLSX files for import, for example. In addition, the backend communicates with Mongoose with MongoDB. |
| **Amine** | To Implement our solution, we worked in agile way for flexibility.  Where we held daily team meetings for collaborations, plus weekly Sprint review to show the progress, identify the risk and append our task list.  We have chosen GIT as Version Control System and GitHub to host our remote repository.    To support each other at all the time and to communicate faster we used WhatsApp and Discord to collaborate. |
| **Edgar** | For the presentation of our program we will go through different use cases to give you a better idea how our program works. At first we show you the connection to the database with credentials. After that we show how to create or load multiple diagrams. There you can see how to create applications and dataflows. Also we show you the properties from the applications, the dataflows and data objects. Next you will see how you can export personal data and data objects as a csv file. Then we show how to import data from excel. At the end we show you how to manage data objects.  **(Go to the program)**  **…** |
| **Ersan** | **Application started**  In the first Screen of our Application, you will be greeted with a form to Connect to a Database. Here you can Fill in the IP-Address with Port and you can also define the specific Database you want to work with. Also, you have to provide a Username and Password for Authentication to the MongoDB database. The latest Information you fill into these field will be stored on the local copy of the Application and will be loaded as default values for a fast workflow.  When you have entered the Connections Information and click on Connect…  **(Auf Connect drücken)**  … you will be directed to the next window, where you can choose whether you create a new Diagram, or Delete or Work with an existing Diagram. In this Case we want to delete an existing Diagram….  **(Demo delete löschen)**  …. and create a new Diagram….  **(Neues Diagram erzeugen mit dem namen: “Demo create“).**  When a Name for the Diagram is set and confirmed with the Button “CREATE”. The diagram will be generated in the Database and will be opened.  But for the following Presentation we have also generated a real-World example to show our other Features we implemented.Therefor you click on the Homebutton and you can select the right diagram  **(Zurück zur Diagramm übersicht und Demo Diagram auswählen).**  **Nächster Präsentierender übernimmt** |
| **Karsten** | **Program Presentation**   * As you can see, this is a simple real life diagram we prepared   **=>(Palette zeigen mit Maus )**   * On the left side you see the palette where you can drag and drop an application on the canvas   **=>( Drag & drop eine Applikation auf den canvas )**   * And you can link each application to another to create a dataflow   **=>( Verlinke 2 Applikations )**   * The user can add now to each to every dataflow multiple data objects   **=>( Auf ein Link gehen mit vielen Data Objects )**   * And every application and data object has their own respective properties as already mentioned before   **=>( Auf ein Link klicken um Properties zu zeigen )**   * The data object properties can be shown here in the data flow * They consist of the name, description and if its personal data or not * By clicking on the data object it will show the description   **=> ( Add and delete ein Data Object )**   * You can add at the bottom a data objects to the data flow or delete on by clicking on the bin symbol   **=>( Auf eine Applikation klicken um Properties zu zeigen )**   * If you click on an application you can see the application properties * The user can fill in a text in the properties: Name, Version and description * COTS is being regulated for the User with a dropdown menu with 3 values * The release and shutdown date accepts only a date format and can also be filled by clicking into a calender and choosing a day * Another gimmick concerning the dates is, that depending on the filled in release and shutdown date, the application will change its color like a traffic light system   **=>( Zeige kurz eine grüne, rote und gelbe Application )**   * Green: if the application is still running and not shut down * Amber: if the release date is in the future * Red: if the shutdown date is already in the past * The program is always comparing those dates to the date today   **Finish** |
| **Jan** | Now let's talk about the two different export functions. When you open Export in the menu, you will see two export options. The exported file will be saved as CSV. If you click on "Export Personal Data" a table will be created. This table contains all programs that work with personal data in the data object. Clicking on it opens a new window. In this window you can choose where on your computer you want to save the file. This file can be opened with Excel or imported into Excel.  The other "Export DataObj" exports all data objects used by the current diagram. Again, you can select the location and open it with Excel. |
| **Amine** | The user can import application into the diagram from an existing excel file.   * At first, he is going to be asked to choose a excel file. * After that the file is going to be temporary uploaded into the server to be analyzed for the first time. * The user is going to be asked to choose a sheet select which column represents which Application property. * This column-Property mapping will be saved first in locally then after that in the remote database, so by the next import the process is easier. * After already uploaded file in the server is going to be analyzed deeply, and all candidate rows are going to be converted into Application. * Rows containing false data, are going to be ignored. * At the end, a report containing all information about ignored Rows/Application is going to be generated and delivered. * To add the Application into the diagram. |
| **Edgar** | **(Go to the sidebar)**  In the sidebar you can also select manage data objects.  **(Manage data objects)**  The user can edit here an existing data object or create a new one.  **(Create “test”)**  If you want to create a new one the user can fill in the properties name and description and can check or uncheck the personal data property.  **(Save)**  Now the dataobject is in the dataobject list so the user can add the new dataobject to any dataflow as you have already seen.  **(Sidebar->Manage->Edit)**  You can also edit existing data objects. You can scroll through the dataobjects list and click on the one you want to edit.  **(Choose one dataobject and edit it)**  You can either delete or change the properties of the dataobject. If you want to save the edited dataobject you can press the save changes button.  **(Save)**  The user will get back to the application landscape and all dataflows where this dataobject was a part of will be updated with the changes. You can always abort and go back to the manage dataobjects menu.  **(Sidebar->Manage->Edit)**  Like before you can choose one dataobject and delete it  **(Choose dataobject “test” and delete it)**  Now you can go back to the manage dataobjects menu  **(Go back to menu)** |
| **Ersan** | **Ende** |