University of New Haven

Tagliatela College of Engineering

Victoria Oliveira
Louis Notarino
Cinthya Grajeda
Kevin C. Gonzalez
Kyle Pinder
University of New Haven
300 Boston Post Road
West Haven, CT 06516

Erik Blair
DocuWare Corporation
35 Thorpe Avenue
Suite 201
Wallingford, CT 06492

June 20, 2016

Dear Mr. Blair,

We hereby submit the report regarding our project entitled "Business Intelligence Dashboard". This report entails all information regarding the approach, development, and implementation of our team's past two semesters of work. Details on the choices that the team made for both the design and testing of the project until the final stage are included. This project was a way for us to learn greatly about creating a web application as well as what it is like to have real experience.

We would like to thank you for taking the time to read this report and for the opportunity that we were given to further our education and to learn irreplaceable skills.

Sincerely,

Victoria Oliveira, Louis Notarino, Cinthya Grajeda, Kevin C. Gonzalez, Kyle Pinder

University of New Haven

Tagliatela College of Engineering

DocuWare
Business Intelligence Dashboard

Submitted to Dr. Frank Breitinger
Team Members: Victoria Oliveira, Louis Notarino, Cinthya Grajeda, Kyle Pinder,
Kevin C. Gonzalez

June 20, 2016

Acknowledgements

We would like to thank our sponsors from DocuWare: Erik Blair, KaYoung Ky, and Jon Langdon. Thank you for allowing us to work with you all in creating a unique and interesting product as well as, granting us access to DocuWare's REST API and an extended free trial of your product.

Also, a very special thank you to Dr. Alice Fischer and Dr. Frank Breitinger for providing us with advice and guiding us to our goal.

Executive Summary:

The purpose of the Business Intelligence Dashboard is to allow users to create design models using the data they have access to in DocuWare. The system will offer a graphical representation of any filtered search done by the user. The user will be able to compare multiple invoices or documents side by side as well as creating more than one dashboard at a time. The interface will be web-based, and tablet friendly through the browser. Users will have the ability to create, save, print or email charts, and will be able to display them using a tablet or any other portable device.

Upon DocuWare's request, C# was used in the backend, and HTML5, CSS3, and JavaScript in the frontend. The graphs were made possible by HighCharts[1]. Twitter Bootstrap v3.3.6[2] allowed the web-based system to resize appropriately for smaller devices.

Users are able to create graphs out of their stored documents, invoices, etc, easily. Through the left sidebar, a user will select a graph type and then the criteria of their choosing. The graph will generate on the page within a dashboard.

The system will integrate with DocuWare's REST API[3] to gain access to the database server. Without this access, the user would not be able to generate graphs and charts with their stored data. There were no costs needed to create this design, as we utilized free applications. DocuWare's proposal is displayed in Appendix A in this document.

Table of Contents

1. Introduction

Companies around the world have many documents. They are stored in large file cabinets, boxes, large hard disks, and stacks in and on employees' desks. But there comes a time when their storage space is full. DocuWare has developed a system that allows companies to store their documents, invoices, etc, in their own private cloud space.

DocuWare is a worldwide company that developed a web-based Electronic Content Management (ECM) software that is available in 70 countries. Their customers have the ability to send any document to DocuWare's cloud database[4], which is branched off to the customer's own private section. This allows the customer's associates to view the documents with the permission of the owner.

The customers of DocuWare have saved vast amounts of documents, but are unable to create charts and graphs using their own stored data. The proposal DocuWare has given us contains some constraints. The application must be web based, must be able to work on mobile devices such as tablet and smartphone. The user must be able to create graphs and charts of their data. The user must also be able to make multiple dashboards that can be saved. The application must connect to DocuWare's REST API otherwise there won't be any documents to work with and the project would fail.

2. Problem/Project Definition

Docuware desires a software design that is connected to their REST API[3]. The problem is that DocuWare currently does not have the means to allow their customers to be able to create a graphical representation using data from the customer's cloud storage.

DocuWare has given us the task to create a dashboard that can be used on a mobile device, such as a tablet or phone. The purpose of this system is to add a new service layer to DocuWare's already established platform of software and cloud services[4]. This interface will be able to satisfy stakeholders' needs of transforming data into meaningful information that is helpful for their business' success. Users, such as administrators or managers, will be able to transform filtered data search results into graphical elements, such as charts and graphs. Users will be able to customize their own dashboards with the style they choose. Moreover, this system will allow users to save and print reports making the converted information more meaningful for the user. Finally, the system will allow users to view their data in stationary and mobile devices making their experience available in different platforms.

The new interface will save users valuable time that would have been wasted with the old system, for the customer would have to use a separate application to acquire the required results.

2.1 Requirements

DocuWare requested the following for the project:

- Web based
- Tablet/Mobile device friendly
- Connect to DocuWare's REST API
- Languages:
    - HTML5
    - CSS3
    - JavaScript
    - C#/.NET

3. Evaluations of Alternatives

Under the discretion of DocuWare, several iterations and design changes were created throughout the developmental process. These alternatives were a guiding factor towards progression and were imperative to the final result.

3.1 Computer Science Theory

A crucial factor to the creation of the project was being able to apply the theories we had learned in previous computer science courses. One course that was implemented throughout the developmental stages was the Human-Computer Interaction course. This helped us design a smooth, clean looking User Interface. The outline and initial design of the project was influenced by Software Project Analysis and Design. By applying knowledge on mockups, behavioral and structural UML diagrams, the idea of the project could be brought to reality.

3.2 User Interface

When the project began, a total of seven mockups were created in order to produce the ideal model. From there, two of the mockups were chosen to be the used for the final product. These mockups were presented to DocuWare and the pros and cons of each were discussed. The process of creating mockups was accomplished before and after each meeting with DocuWare for the best possible result.

3.3 Page Alignment

One of the most important design changes made to the web application was the use of Bootstrap v3.3.6[2]. Originally, the web page was a blank slate, relying only on "@media" queries to resize the page based on the pixel size of the screen. That alone became a daunting task where there as so many different size devices and no standard. Bootstrap v3.3.6 made creating a dashboard that could be implemented on a small scale devices easier on the group. By utilizing rows and columns, Bootstrap keeps the page aligned and dynamic so that no matter the screen size, the page will be loaded properly and resized accordingly.

3.4 Left Sidebar

The left sidebar on the main page was a design that kept changing constantly throughout the project. The issue with its first iteration was that it would overlap content on the page and not resize with the window. It was static. So we decided to work on making the sidebar a push menu instead of a slide sidebar. The push menu would push on the screen, with the help of bootstrap, and resize the screen that was pushed over. This fixed the overlap issue.

4. Design Approach

DocuWare requested that the web-based system must be small-scale device friendly, and it must connect to their API. Their requirements helped the group focus on the next steps rather than trying to determine what languages to use.

The group had to focus on developing Use Cases that would fit well with DocuWare's proposal. We came up with three distinct Use Cases: login, generating graphical dashboards, and saving the dashboards. These Use Cases define major parts a user must be able to do with this system. Once these were made up and sound, we moved on to the Class Diagrams. These diagrams represented each Use Case and a high level diagram. Then we created State Diagrams to help visualize the flow of the system. Like the Class Diagrams, we created one for each Use Case and then an overall one.

The group then focused on creating a mobile friendly user interface (UI). The group designed many mockups of how the main page should look, along with designs of a login page, a graph wizard, and the final product of the creation of the graph. We then chose a few designs that we felt best represented what DocuWare was looking for. However, we needed to recreate the mockups of the main page after consulting with DocuWare.

The login page design was simple and a played crucial part in testing. This was the portal, the ultimate connection between our interface and DocuWare's interface. Even though DocuWare already has a login page, we needed our own in order to achieve a connection to DocuWare's existing system through their REST API. This connection was a pivotal point because if we could not connect, then we would not be able to access any of our populated file cabinet. Without a populated file cabinet, we would not be able to create graphs with the correct information.

5. Design Narrative

For the login page, we went with a simple login design, we used DocuWare's logo and color scheme that is currently being used in their main website. We retrieved the colors, logo, and clouds in real-time from DocuWare's website. It also also has the same login format DocuWare has which is the name of the company, username, and password. We also made the DocuWare logo clickable so it will redirect the user to the DocuWare website. We started with the login page because it was a good start for us to take since we needed to make sure that we can login to our cloud database in DocuWare. This page played a huge role in testing as it was critical to be able to access the data in our accounts.Once we were able to successfully connect to our cloud database, we went ahead and started designing the main page.

5.1 Use Cases

Three Use Cases were thought up and planned out to help layout the important aspects of the system. The use cases are: log in, generating graphs, and saving graphical dashboards. Below are the flows for each of the cases.

5.1.1 Use Case: Log in

Main Flow:

1. The user enters all credentials correctly, username and password.
2. The system verifies that all credentials are valid.
3. The system allows the user access to the web-interface.
4. Main flow ends.

Alternate Flow 1:

1. The user enters some, or all, credentials incorrectly.
2. The system notifies the user that the credentials are wrong.
3. The user tries to log in again and succeeds.
4. Alternative flow ends.

Alternate Flow 2:

1. The user knows their user-name/E-mail but forgets their password.
2. The user clicks on "Forgot Password". They receive an E-mail with their password/randomized password and is able to log in with those credentials.
3. Alternative flow ends

Alternate Flow 3:

1. The user knows his password but forgets user-name/E-mail. The user clicks on "Forgot Username".
2. The user is sent to a page to answer security questions and retrieve his username.
3. The user answers the questions successfully and is able to go back and log in.
4. Alternative flow ends.

Alternate Flow 4:

1. The user is a new user and wants to try the free trial.
2. The user signs up for the trial and completes the sign up process.

3. The user confirms account via email and is granted access.
4. Alternative flow ends.

Alternate Flow 5:

1. The user is a new user and wants to try the free trial.
2. The user signs up for the trial and completes the sign up process.
3. The user confirms account via email and is granted access.
4. Alternative flow ends.

Exception Flow:

1. The user enters some, or all, credentials incorrectly more than the set limit of 5 times.
2. The user is locked out from trying to log in again and notified to call customer service.
3. The use case fails.

5.1.2 Use Case: Generating Graphs

Main Flow:

1. The user filters the required data from the database of invoices.
2. The user clicks on the graph wizard and chooses a bar graph.
3. The user is given the option to name the X-axis and Y-axis.
4. The user prompts the program to calculate the total cost of all invoices by month.
5. The user assigns the total cost data results to the Y-axis portion of the graph.
6. The user assigns the monthly portion to the X-axis.
7. The user exits the wizard.
8. The graph is created showing the total cost per month of all invoices.

Alternate Flows:

1. The user fails to create the desired graph
2. The system notifies the user the data is insufficient, unmatchable or invalid and to try to create the graph again.

Exception Flows:

1. The user fails to create the desired graph with insufficient and unmatchable data.
2. The user is notified to click on the help button, tutorial guide or contact IT for further help.

5.1.3 Use Case: Saving Graphical Dashboard

Main Flow:

1. The user selects the graphical dashboard/s they want to save to disk
2. The user clicks on the Save As button.
3. The user selects the format they desires.
4. The user saves the graphical dashboard.

Alternate Flows:

1. The source location to save the file is invalid.
2. The system notifies the user to verify the location exits and try again.

Exception Flows: Errors generated for incorrect export path.

1. The user is notified the location still invalid and is notified to log out of the system and try again.
2. The employee logs out and verifies the location folder exits and the use case ends in failure.

5.2 Class Diagrams

Multiple class diagrams were created in order to provide structure when creating the program. Figure 1 shows the High Level Class Diagram that was used during the project's creation. The diagram models an overview of the data used within the software.

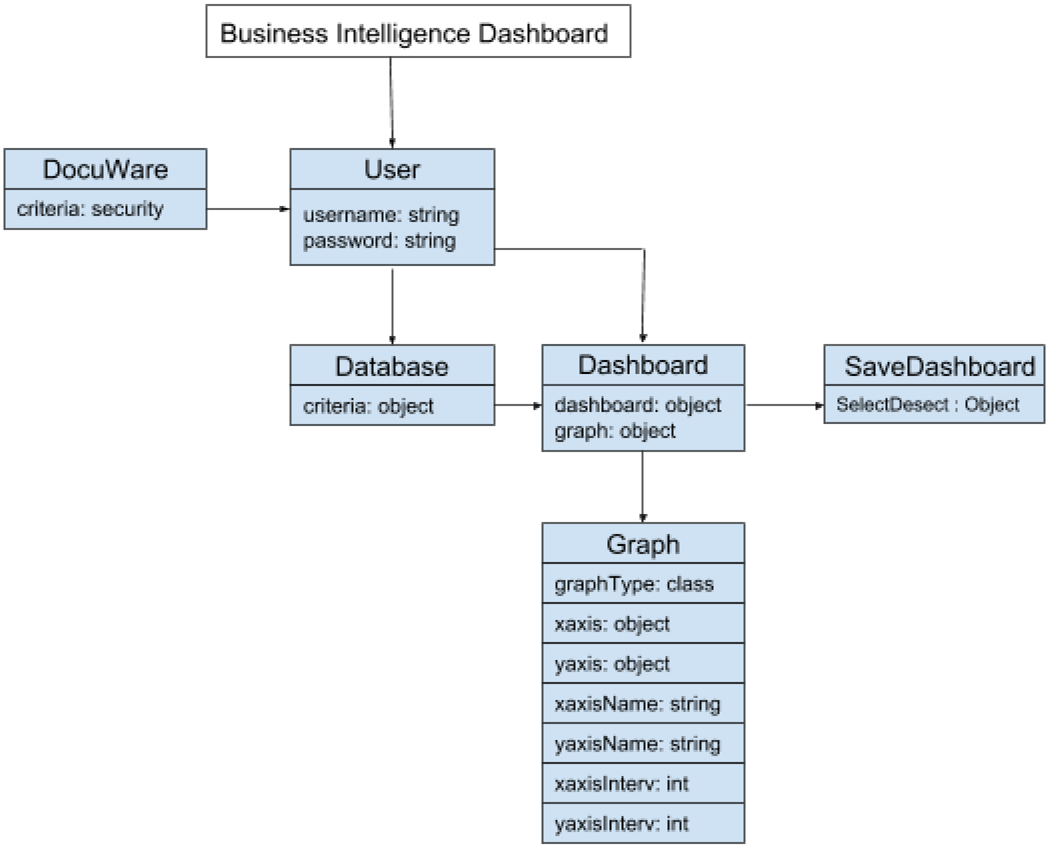


Figure 1: High Level Class Diagram

Figures 2-3 are models of the data for the Login and Generating Graphical Dashboards. In the Login class diagram, the User class initiates the login operation by entering a username and password, then the Docuware class initiates both of its operations to retrieve the username and password and goes through security to check the values. In the Generating Graphical Dashboards class diagram, the Dashboard class contains the functions to create and delete dashboards, search data, remove graphs, and a copy/paste function. The Graph class contains several attributes in order to make graph.

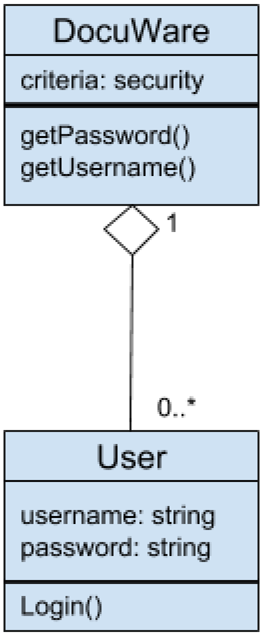


Figure 2: Login Class Diagram

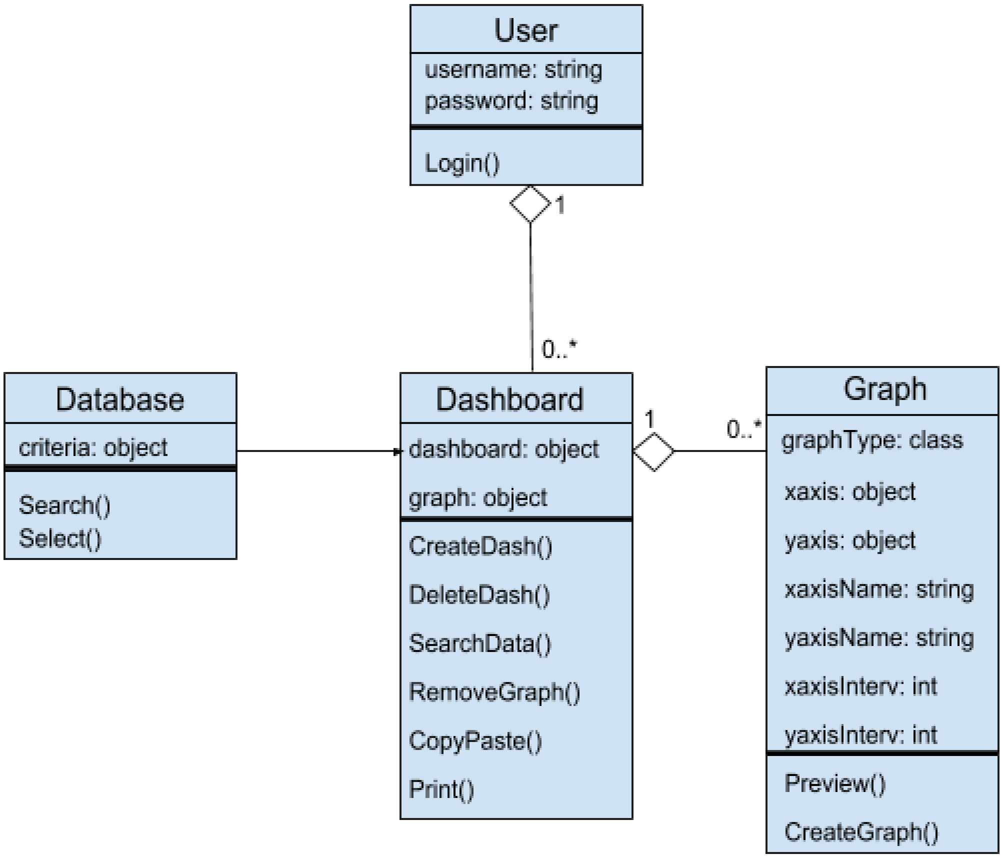


Figure 3: Generating graphical dashboards Class Diagram

Figure 4 is the model for the Saving Graphical Dashboard class diagram. The diagram shows the SaveDashboard class which contains the functions to select a dashboard and save it.

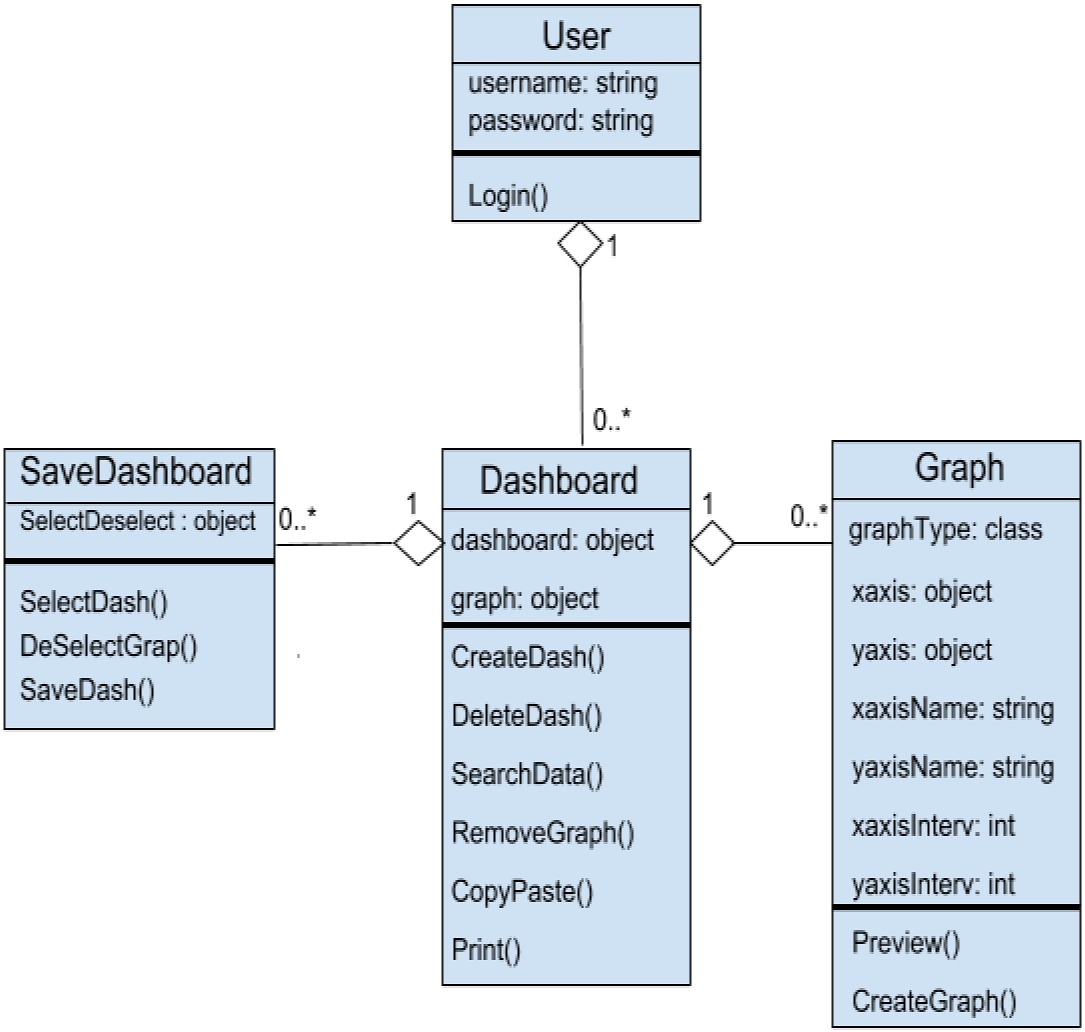


Figure 4: Saving the graphical dashboard Class Diagram

5.3 State Diagrams

State diagrams were created alongside the class diagrams in order to illustrate the functionality of the program. The first state diagram, figure 5, illustrates the path that the system takes and any alternatives that may occur. Following the overall state diagram are the state diagrams for the three use cases. Figure 6 demonstrates the states of the login page, figure 7 shows the possible states when a graph is generated, and figure 8 shows when a dashboard is being saved.

When the program is started, the user is prompted to enter their username and password. Once the account has been verified, the user is connected to the system and can create a dashboard. If the login fails, the user is prompted to re-enter their credentials. If it fails a second time, the system ends. If the user is able to create a dashboard, they can then generate graphs and save their data. Any time after they are connected to the system the user can log off and disconnect from the program.

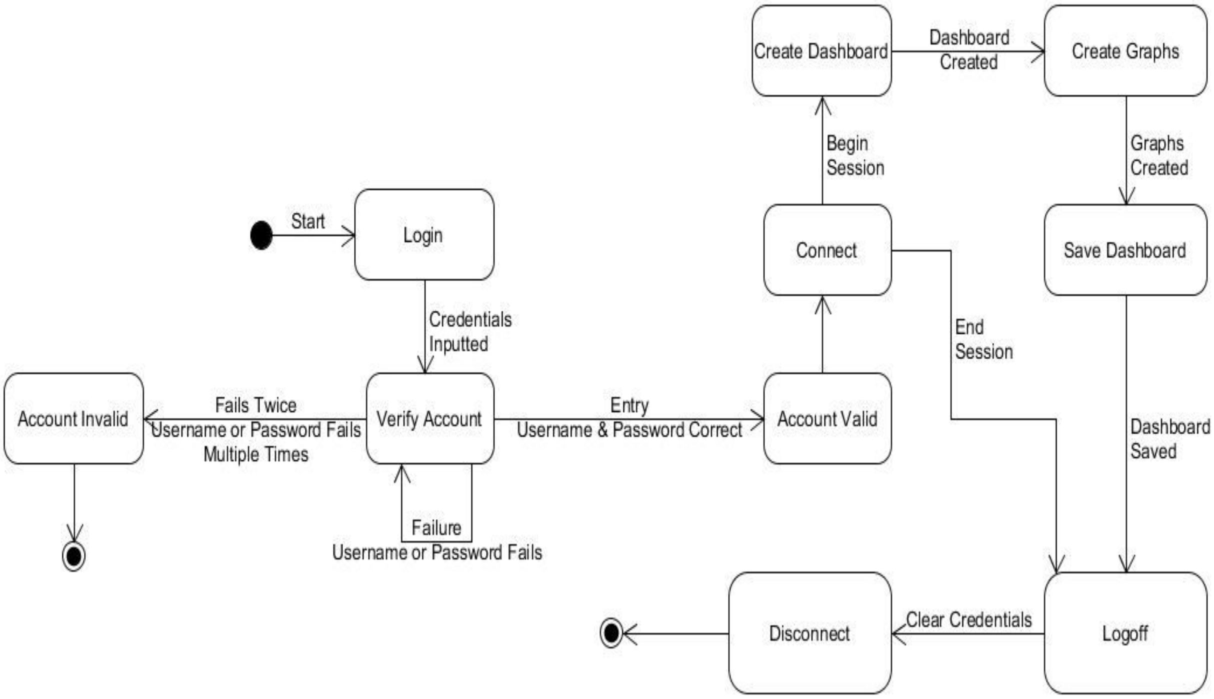


Figure 5: Overall State Diagram representation of the system and its Use Cases

Figure 6 provides greater detail about the login procedure. When the user is asked to enter their login information one of two possible results occur; either the credentials are verified or they fail. If the process fails twice, the account is locked and the program ends. When access is granted the session begins. The user is then able to gain access to the main page before logging out.

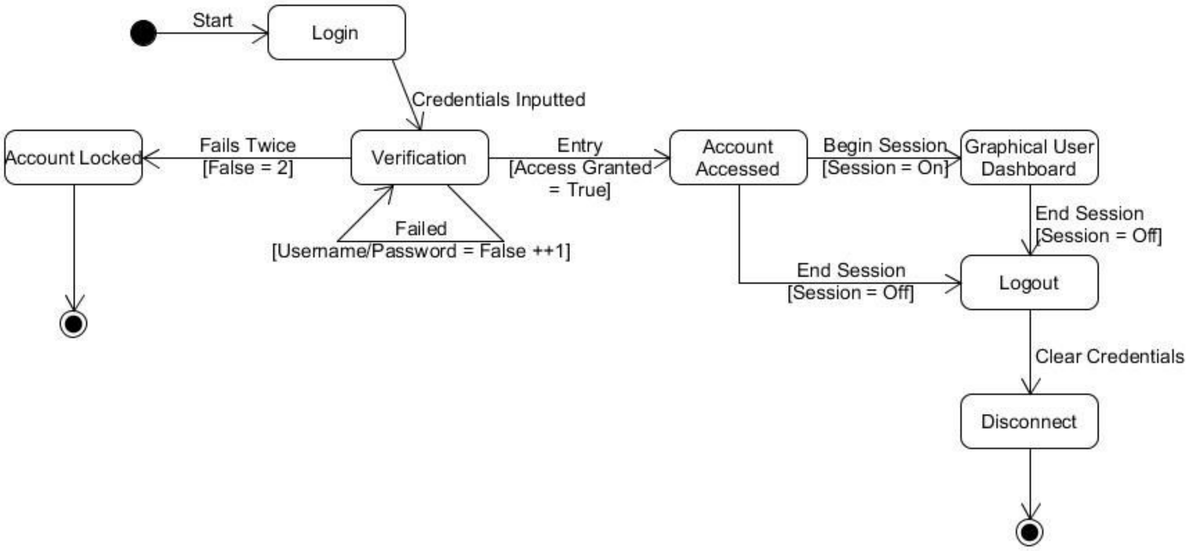


Figure 6: Login State Diagram

When the user has logged in to the system they are able to end their session at any point in time. Once on the main page the user can create a dashboard and filter their results. After the data is selected the user can create as many graphs as they would like. Figure 7 focuses on the details of creating dashboards and graphs.

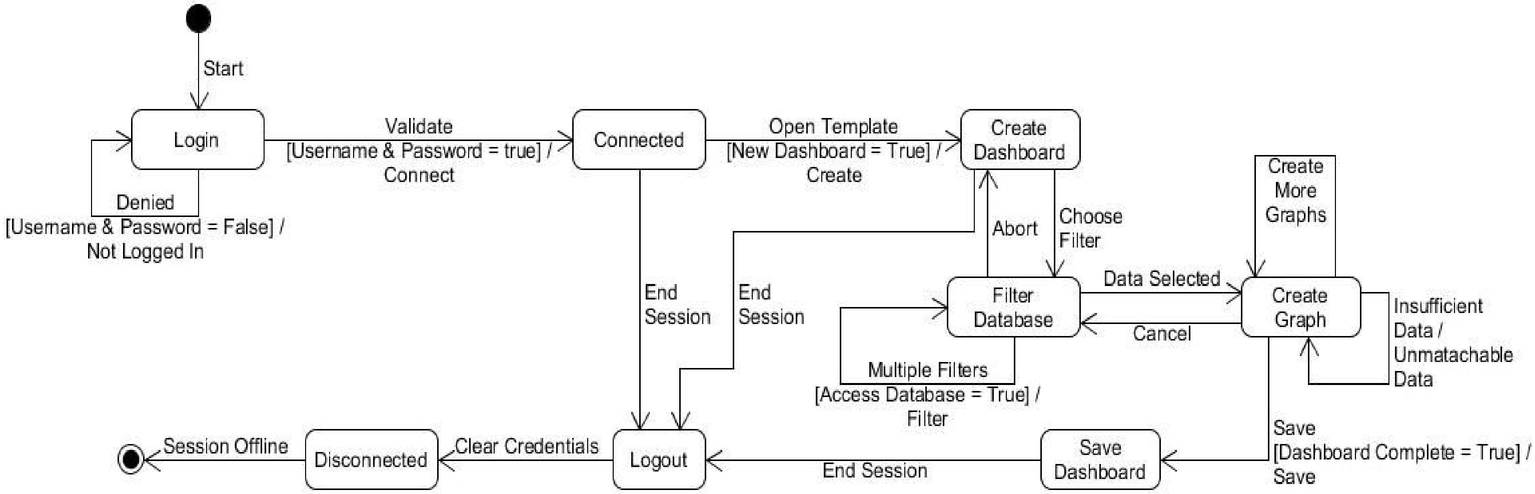


Figure 7: Generating graphical dashboards State Diagram

Figure 8 details the final steps of the system. After the dashboards and graphs have been created, the user can save the visual data that they want to access later. When the save state is validated the user will have their information stored in .pdf format. They are able to repeat this process as many times as they would like until they choose to log out.

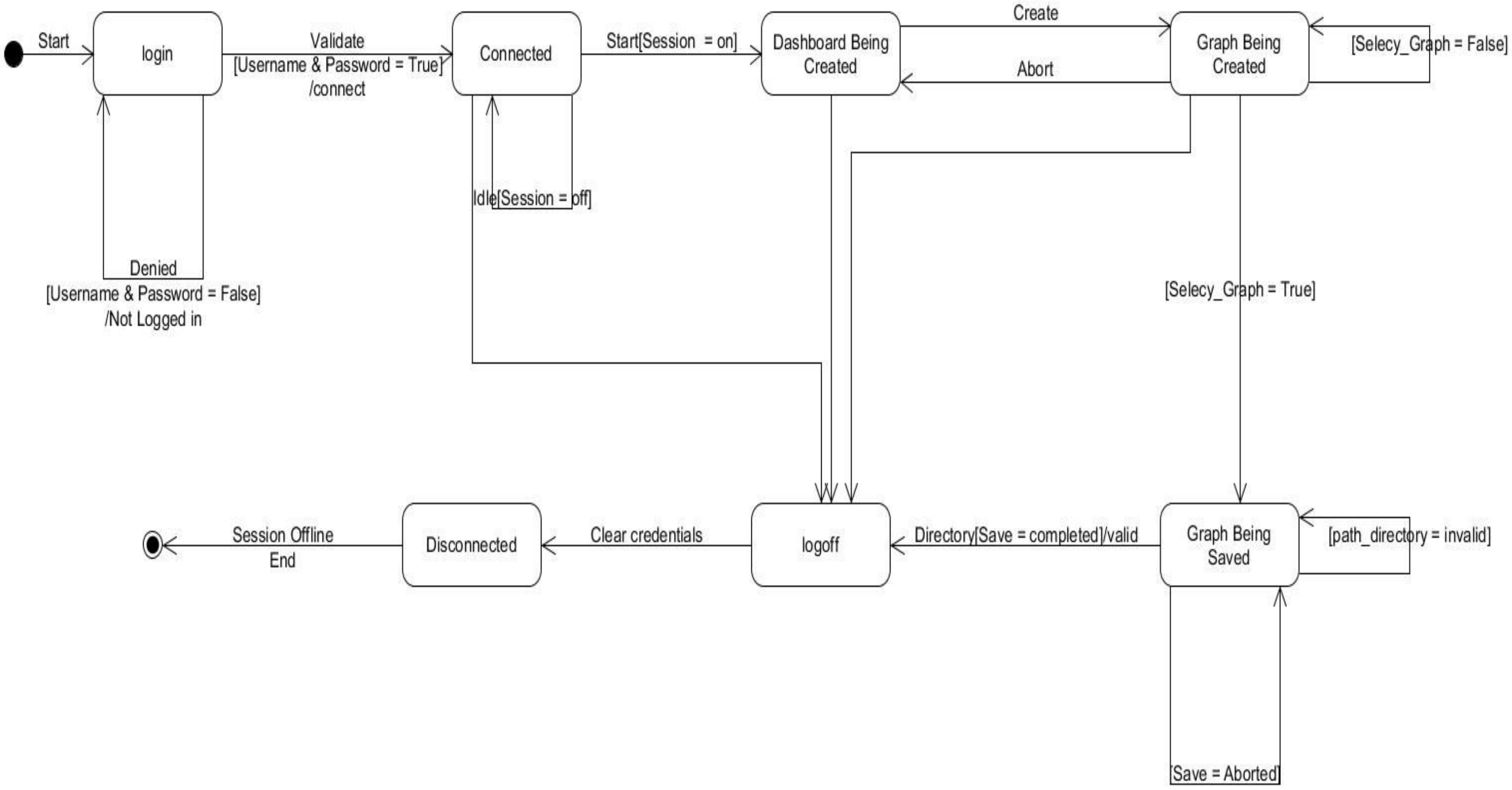


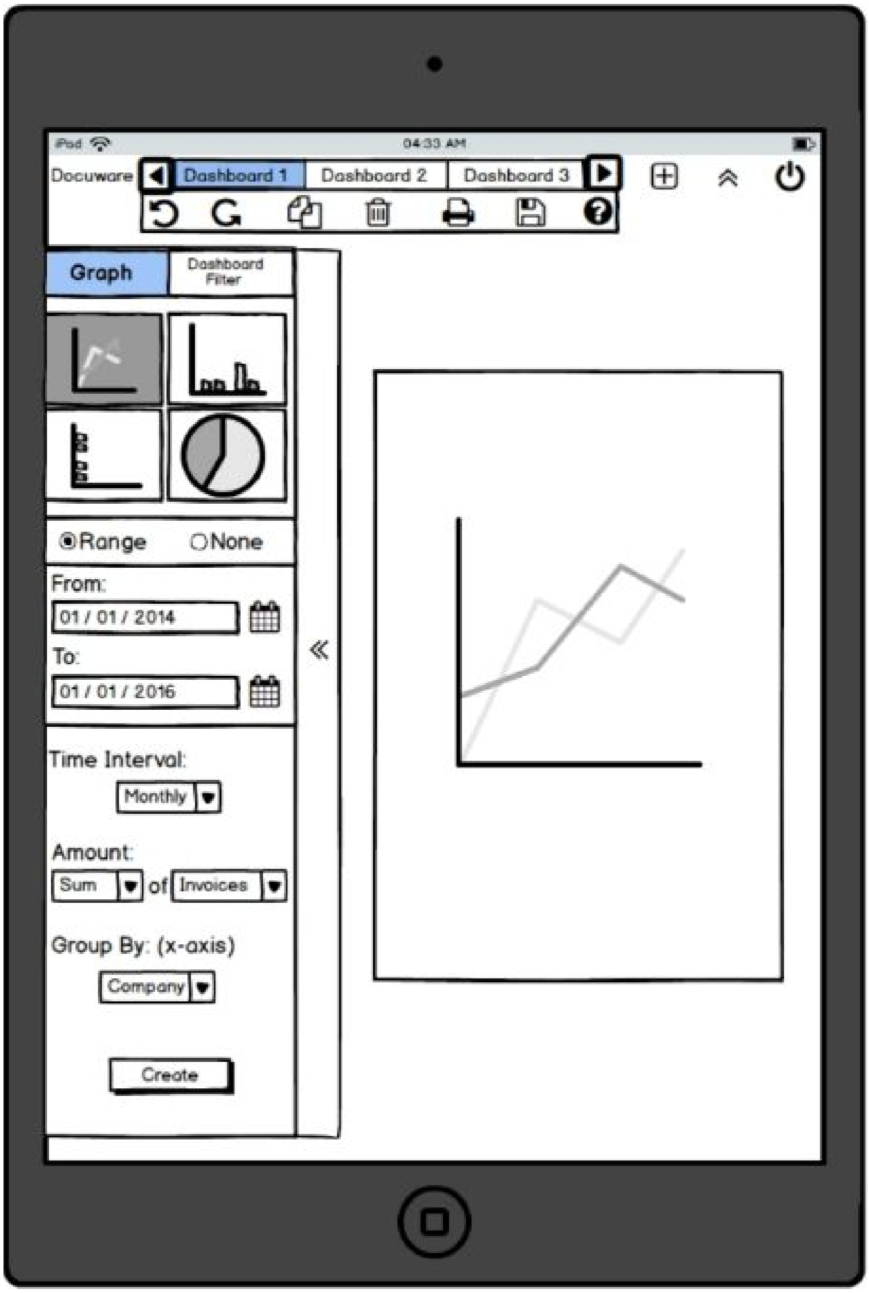
Figure 8: Saving graphical dashboards State Diagram

5.4 Mockups

After the Use Cases, Class Diagrams, and State Diagrams were completed, the group was then able to design the User Interface for the system. The diagrams and Use Cases were helpful during the early mockup versions. MyBalsamiq allowed the group to design the mockups that are shown from Figures 9 to 11, which were approved by DocuWare.

For the main page design, we created a toolbar which houses a few tools such as a save, print, and delete which is hidden underneath the dashboard icons. We also included a sidebar which houses the graph wizard and dashboard filter.

Below, Figure 9 shows a final mockup of the main page with a full view of both menus with a graph on the page. There is a static top navigation bar that contains the different dashboards. On the right side of the top are buttons: the plus will add a new dashboard, the double down arrow button will drop down a toolbar that is under the navigation bar, the last button is the logout button. On the left side of the navigation bar we added the DocuWare logo for visual purposes. The left sidebar is open displaying the four (4) types of graphs a user can choose from, along with the graph criteria options. On the top is the top navigation bar, displaying the dashboards and the drop down tool bar. This mockup was designed in order to show how the page would look on a tablet with both the sidebar and top toolbar open.



● Range ○ None

From: 01 / 01 / 2014

To: 01 / 01 / 2016

Time Interval: Monthly

Amount: Sum of Invoices

Group By: (x-axis) Company

Figure 9: Mockup displaying both the side and tool bar open with a graph on a tablet.

Below, Figure 10 displays a final version mockup of how the main page would look on a tablet with both the sidebar and top toolbar closed. The graph enlarges and takes up as much space as it can on the page.

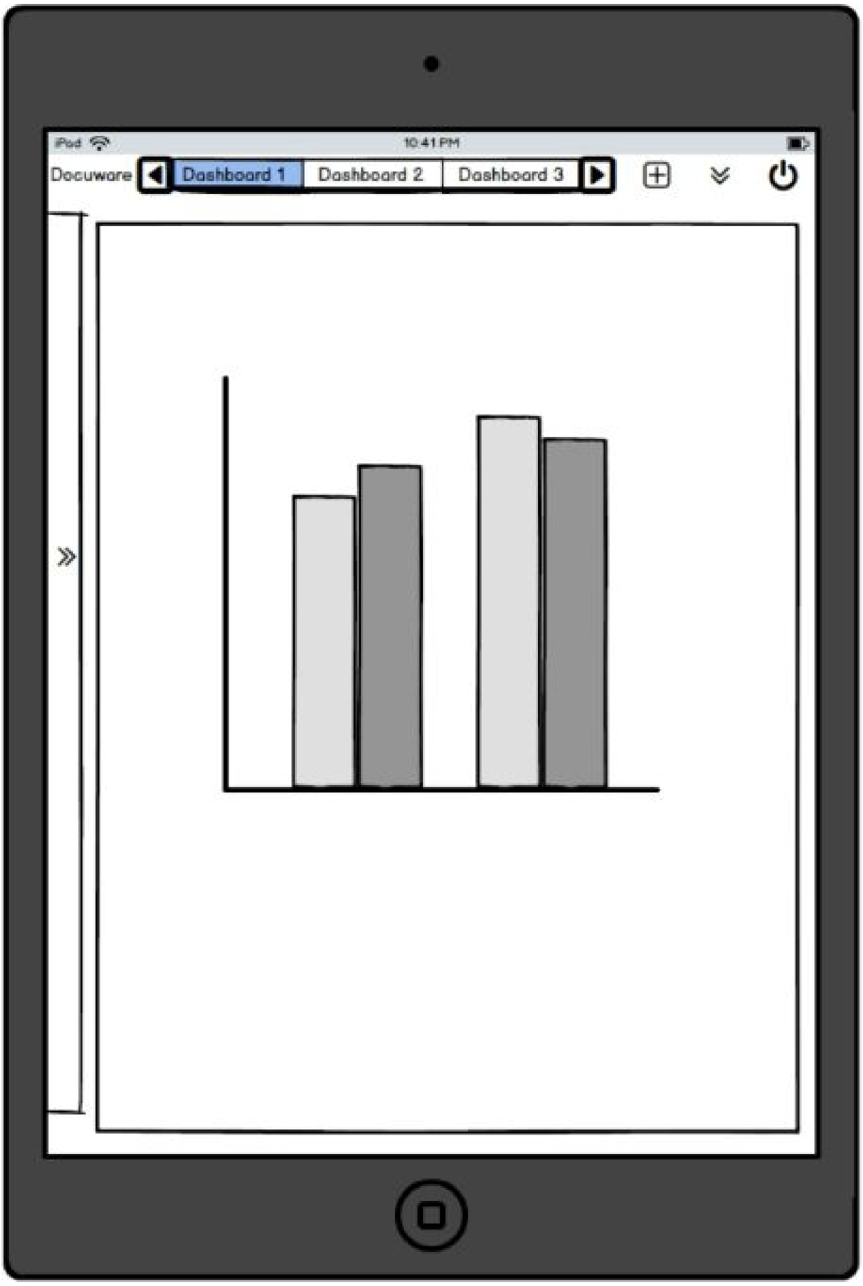
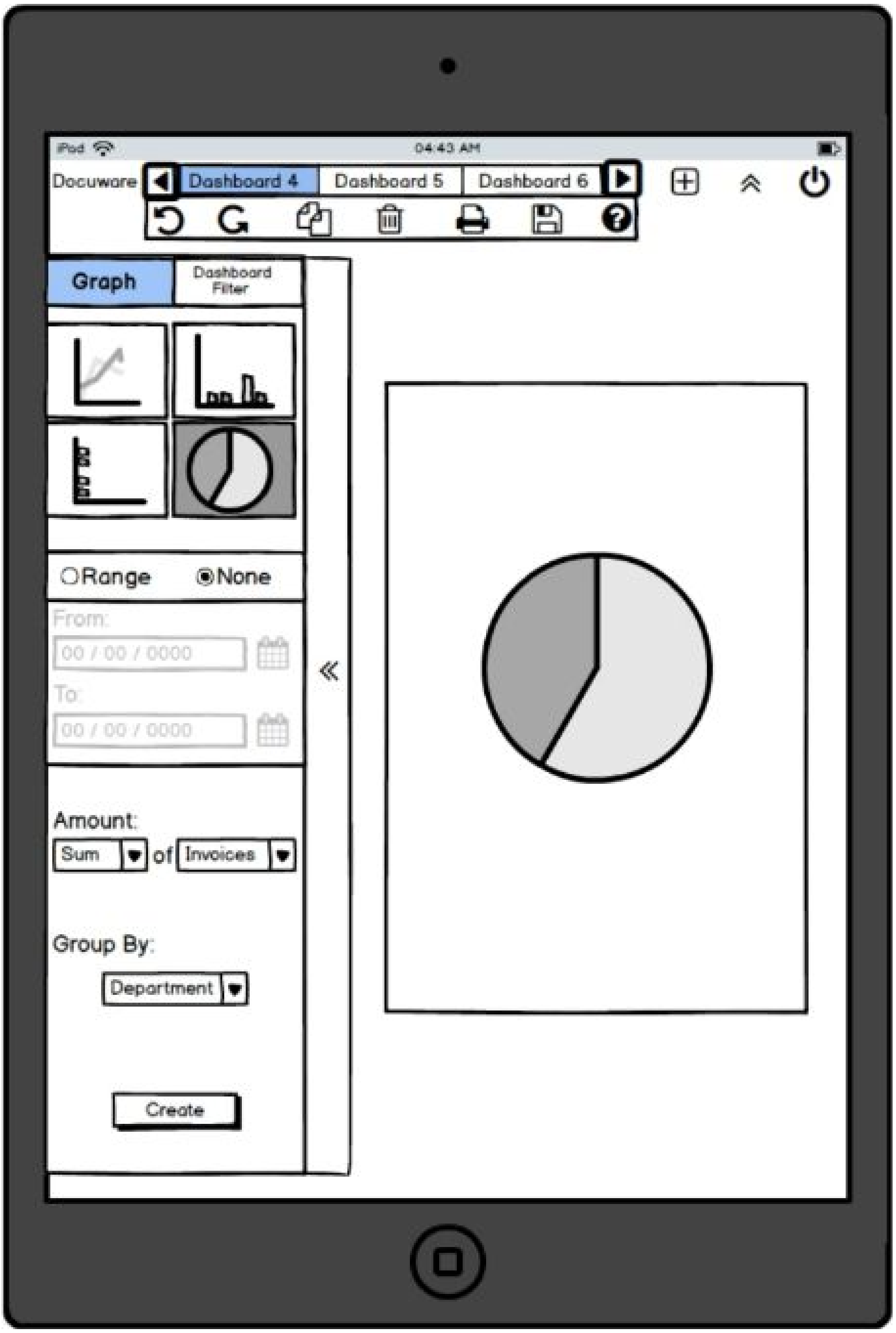


Figure 10: Mockup displaying a graph and both the sidebar and the tool bar closed on a tablet.

In Figure 11, another final version of a mockup of a different graph is displayed with both the sidebar and the top toolbar open. Parts of this mockup have already been described above for Figure 9. This particular graph when selected will grey out criteria that does not apply to the graph, such as a date range.



○Range ●None

From: 00 / 00 / 0000

To: 00 / 00 / 0000

Amount: Sum of Invoices

Group By: Department

Figure 11: Tablet view displaying a pie graph with criteria in the sidebar greyed out to show the options that would be available when a user selects a pie graph

6. Testing

The system has undergone quite a bit of testing. Though there was no true UI testing by outside users, the login and main pages went through many changes based on the testing done by the team members, as well as the feedback received from DocuWare. The file cabinet we had access to was empty, so we had to populate it. We came up with an idea to create five fictional companies, each with the same four departments. This required us to create 500 documents. Each department sent the company one invoice each month, for two years: Jan. 2014 to Jan. 2016.

6.1 Login Page:

This page was the first to be completely built. While trying to connect to DocuWare's REST API, the page was under heavy use. This became debugging time for the login page where data members became lost or the connect to the REST API was not correct. This page faced an issue with resizing; the clouds would disappear after a certain window size. But that was fixed rather quickly.

Even though this page was the first to be fully created and functional, it was the last page to get its Bootstrap upgrade. During the conversion, the clouds became an issue. They would move around as they pleased or disappear when the window was made smaller. But once it was fixed, the issues on this page disappeared. Figure 12 shows the completed look of the login page.

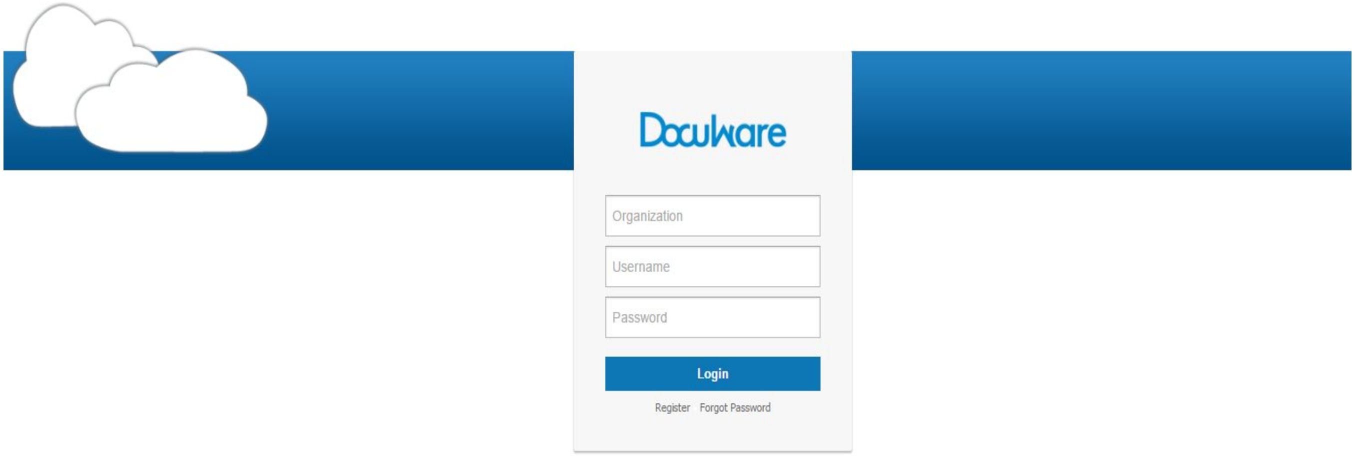


Figure 12: Final version of the login screen

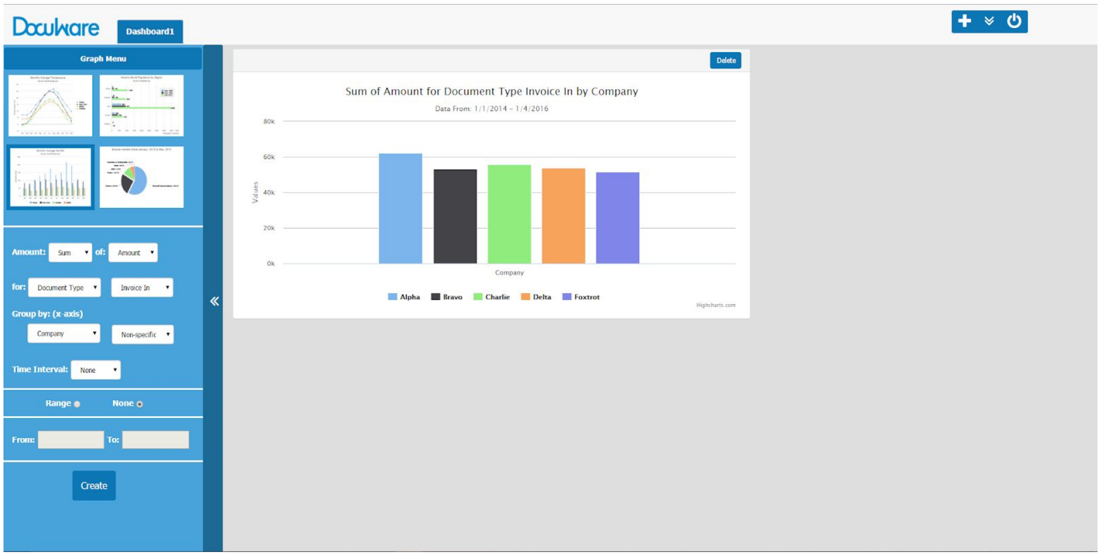
6.2 Main Page:

This page has undergone many revisions, especially to specific pieces. One of its biggest issues was that the sidebar would overlap onto the top navigation bar and page content. This is a huge issue for a project like this. The user wants to be able to see everything, not cover it up while trying to create another graph to represent comparative data. This was fixed and now the sidebar pushes on the page instead of laying ontop of it. Another issue was the tool bar. Like the sidebar, it would overlap onto the page content. At a different time, it was hard to get the tool bar to move by the push of a button.

Controversies on the page layout was another issue that needed to be put to the test. The sidebar button remained on the side and ran down to the bottom of the page. The button for the tool bar to slide out was put on the navigation bar, instead of underneath it running the length of the screen (this was how it was depicted in a first draft mockup). The logout button was determined to be in the top right corner of the page.

The graphs and dashboards needed the most testing. Once HighCharts was functioning on the page, it still had issues within the dashboards. The graphs generated on the page and were functioning properly, however they caused issues with the dashboard generation. If a dashboard was generated and then a graph, the dashboard would disappear. If the graph was made first, and then a dashboard was created, the graph would display on the next dashboard, as well as the one after that and so on and so forth. The graphs were not contained to that one dashboard.

The page re-scales itself perfectly for a user on any size device. This test was performed many times to make sure graphs were not disappearing, along with other items especially on the sidebar. The figure below shows the finished version of the UI for the main page.



Amount: Sum of: Amount

for: Document Type Invoice In

Group by: (x-axis) Company Non-specific

Time Interval: None

Range ● None ○

From: To:

Figure 13: Final version of the main page displaying a single graph

7. Discussion

The purpose of this web application is for a user to view a graph of their desired data. The application is made for portable devices as well as large desktop systems. The end result is something that makes DocuWare's system more efficient and simple to use. It fits on any size screen and can pull data directly from the information fields on an uploaded document. By being user customizable, the graphs that are developed within each dashboard can be made of many different types of data: from the number of documents in a company to the sum of all invoices for the past 'x' number of years. The design of the system allows for an increase in productivity because of the time and energy that can be saved by every user. However, the system does not allow users to make payments on invoices or to search for specific documents. It is only used to view a graph of the data in data fields. The needs and details of the project were established during each meeting with DocuWare. Each component of the web application was reviewed and critiqued until a proper software application that met every goal and requirement was developed. DocuWare now has a successful web application that they can implement into their system; a system that is used around the globe by over 14,000 companies.

8. Conclusions and Recommendations

In conclusion, the final product is something that will change how DocuWare system users work and view their data. What would normally take hours upon hours of data comparison will now be as simple as clicking a few buttons. This web application features user customizability through a simple filter feature and the ability to create and name multiple dashboards. The data used for the graphs is pulled directly from the data fields of each document in the file cabinet. After the system was created it was tested through the creation and implementation of five hundred new files onto a DocuWare account. By using Bootstrap the page can fit on any screen, from a large monitor to a small handheld device. Ease of use is granted further by the retractable toolbar on the top of the screen and the filter on the left side of the screen.

For the future it would be ideal to add more features to the product. Although the web application is in working condition, it takes time to pull information from the five hundred documents that were created for testing purposes. While this will work well for a company that does not have many files on DocuWare's system, a large company with hundreds of thousands of documents will completely slow the web application down to a crawl when they attempt to generate graphs. Next, changes to the dashboards would help the user's experience. Rather than having the first dashboard be open when the web application is loaded, it would be more beneficial for the user to create and name their first dashboard.

References

[1] Highcharts [Online]. Available: http://www.highcharts.com/docs. June 2016

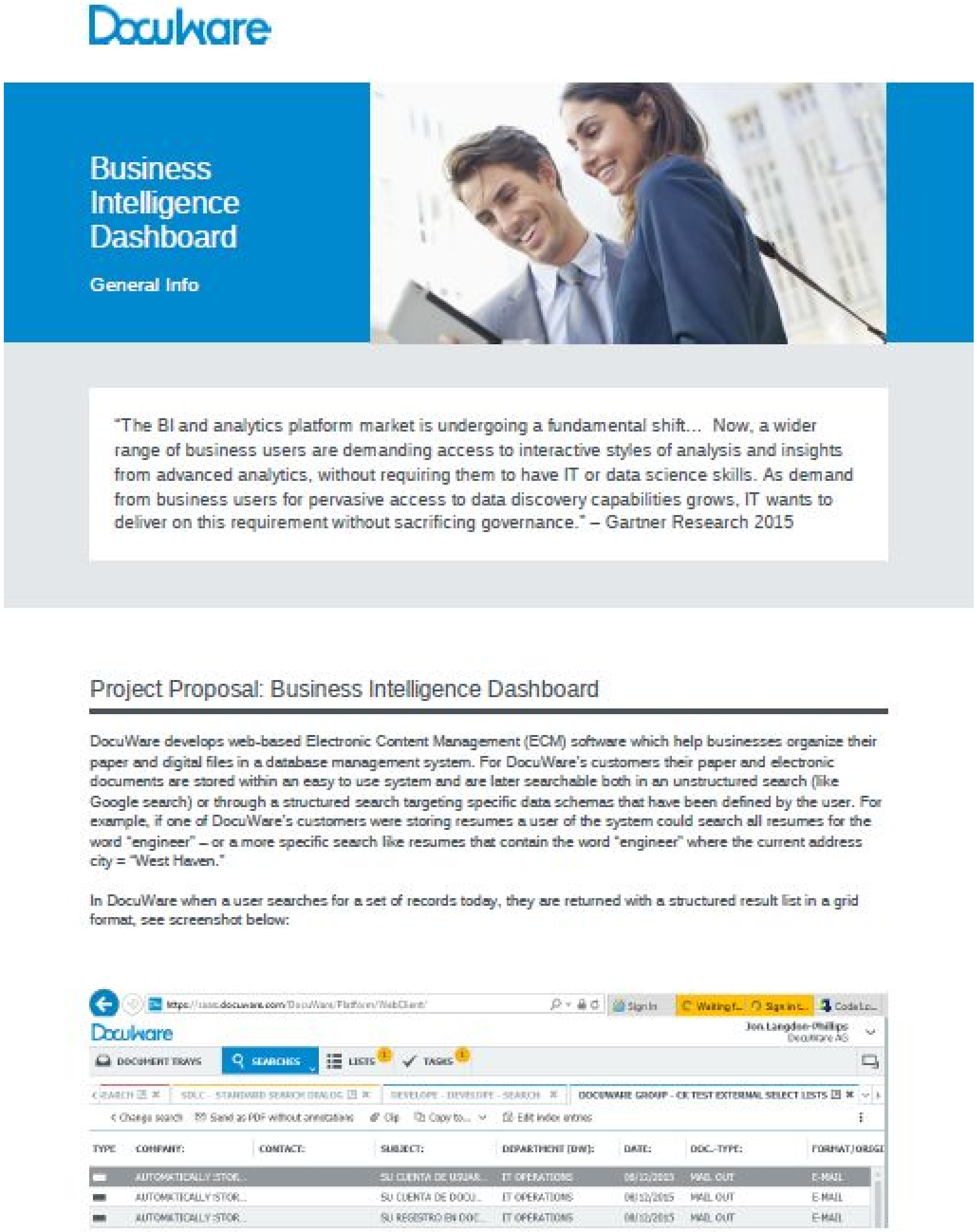
[2] Bootstrap [Online]. Available: http://getbootstrap.com/css/. June 2016

[3] DocuWare Platform .NET API [Online]. Available: http://help.docuware.com/sdk/platform/html/66b2ed1e-2aef-452a-97cd-5014bbf0242B.htm. June 2016

[4] DocuWare [Online]. Available: https://www.docuware.com/document-management-products-and-services/docuware-Cloud. June 2016

[5] myBalsamiq [Online]. Available: https://www.mybalsamiq.com/. June 2016

Appendix A: Proposal



DocuWare

Business Intelligence Dashboard

General Info

> "The BI and analytics platform market is undergoing a fundamental shift... Now, a wider range of business users are demanding access to interactive styles of analysis and insights from advanced analytics, without requiring them to have IT or data science skills. As demand from business users for pervasive access to data discovery capabilities grows, IT wants to deliver on this requirement without sacrificing governance." – Gartner Research 2015

Project Proposal: Business Intelligence Dashboard

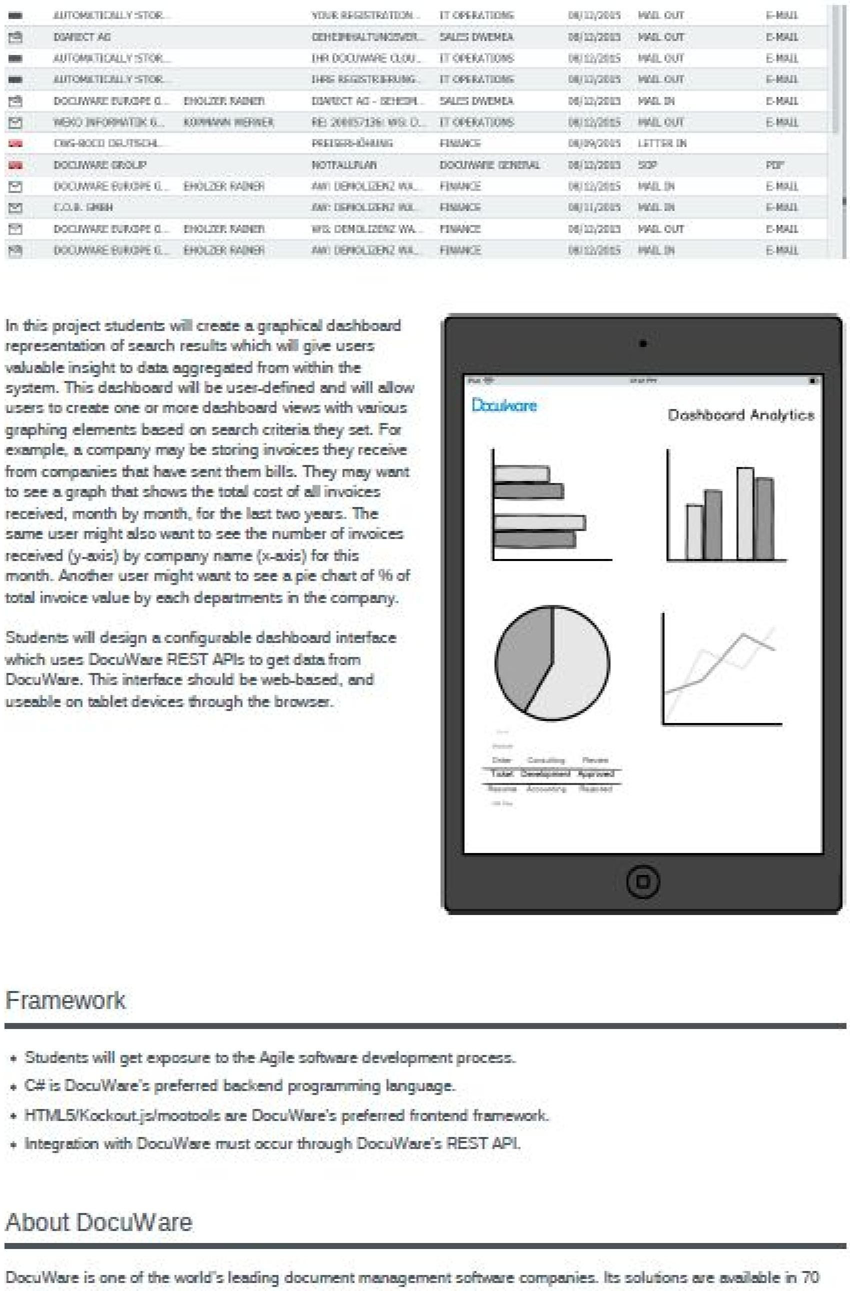
DocuWare develops web-based Electronic Content Management (ECM) software which help businesses organize their paper and digital files in a database management system. For DocuWare's customers their paper and electronic documents are stored within an easy to use system and are later searchable both in an unstructured search (like Google search) or through a structured search targeting specific data schemas that have been defined by the user. For example, if one of DocuWare's customers were storing resumes a user of the system could search all resumes for the word "engineer" – or a more specific search like resumes that contain the word "engineer" where the current address city = "West Haven."

In DocuWare when a user searches for a set of records today, they are returned with a structured result list in a grid format, see screenshot below:

https://saas.docuware.com/DocuWare/Platform/WebClient/
Sign In
Waiting f...
Sign in c...
Code Lo...
DocuWare
Jon Langdon-Phillips
DocuWare AG
DOCUMENT TRAYS
SEARCHES
LISTS
TASKS
SEARCH
SDLC - STANDARD SEARCH DIALOG
DEVELOPE - DEVELOPE - SEARCH
DOCUWARE GROUP - CE TEST EXTERNAL SELECT LISTS
Change search
Send as PDF without annotations
Clip
Copy to...
Edit index entries

TYPE	COMPANY:	CONTACT:	SUBJECT:	DEPARTMENT (DW):	DATE:	DOC.-TYPE:	FORMAT/ORIGI
	AUTOMATICALLY STOR...		SU CUENTA DE USUAR...	IT OPERATIONS	08/12/2015	MAIL OUT	E-MAIL
	AUTOMATICALLY STOR...		SU CUENTA DE DOCU...	IT OPERATIONS	08/12/2015	MAIL OUT	E-MAIL
	AUTOMATICALLY STOR...		SU REGESTRO EN DOC...	IT OPERATIONS	08/12/2015	MAIL OUT	E-MAIL

Figure 14: Page 1 of the DocuWare Proposal



▬	AUTOMATICALLY STOR...		YOUR REGISTRATION ...	IT OPERATIONS	08/12/2015	MAIL OUT	E-MAIL
🖂	DIARECT AG		GEHEIMHALTUNGSVER...	SALES DWEMEA	08/12/2015	MAIL OUT	E-MAIL
▬	AUTOMATICALLY STOR...		IHR DOCUWARE CLOU...	IT OPERATIONS	08/12/2015	MAIL OUT	E-MAIL
▬	AUTOMATICALLY STOR...		IHRE REGISTRIERUNG...	IT OPERATIONS	08/12/2015	MAIL OUT	E-MAIL
🖂	DOCUWARE EUROPE G...	EHOLZER RAINER	DIARECT AG - GEHEIM...	SALES DWEMEA	08/12/2015	MAIL IN	E-MAIL
🖂	WEKO INFORMATIK G...	KORMANN WERNER	RE: 20085713K WG: D...	IT OPERATIONS	08/12/2015	MAIL OUT	E-MAIL
▬	CWG-BOCO DEUTSCHL...		PREISERHÖHUNG	FINANCE	08/09/2015	LETTER IN	
▬	DOCUWARE GROUP		NOTFALLPLAN	DOCUWARE GENERAL	08/12/2015	SOP	PDF
🖂	DOCUWARE EUROPE G...	EHOLZER RAINER	AW: DEMOLIZENZ WA...	FINANCE	08/12/2015	MAIL IN	E-MAIL
🖂	C.O.B. GMBH		AW: DEMOLIZENZ WA...	FINANCE	08/11/2015	MAIL IN	E-MAIL
🖂	DOCUWARE EUROPE G...	EHOLZER RAINER	WG: DEMOLIZENZ WA...	FINANCE	08/12/2015	MAIL OUT	E-MAIL
🖂	DOCUWARE EUROPE G...	EHOLZER RAINER	AW: DEMOLIZENZ WA...	FINANCE	08/12/2015	MAIL IN	E-MAIL

In this project students will create a graphical dashboard representation of search results which will give users valuable insight to data aggregated from within the system. This dashboard will be user-defined and will allow users to create one or more dashboard views with various graphing elements based on search criteria they set. For example, a company may be storing invoices they receive from companies that have sent them bills. They may want to see a graph that shows the total cost of all invoices received, month by month, for the last two years. The same user might also want to see the number of invoices received (y-axis) by company name (x-axis) for this month. Another user might want to see a pie chart of % of total invoice value by each departments in the company.

Students will design a configurable dashboard interface which uses DocuWare REST APIs to get data from DocuWare. This interface should be web-based, and useable on tablet devices through the browser.

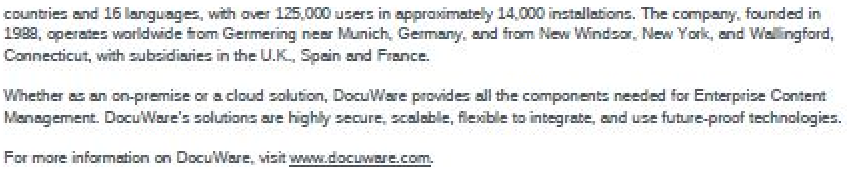
Framework

- Students will get exposure to the Agile software development process.
- C# is DocuWare's preferred backend programming language.
- HTML5/Kockout.js/mootools are DocuWare's preferred frontend framework.
- Integration with DocuWare must occur through DocuWare's REST API.

About DocuWare

DocuWare is one of the world's leading document management software companies. Its solutions are available in 70

Figure 15: Page 2 of the DocuWare proposal



countries and 16 languages, with over 125,000 users in approximately 14,000 installations. The company, founded in 1988, operates worldwide from Germering near Munich, Germany, and from New Windsor, New York, and Wallingford, Connecticut, with subsidiaries in the U.K., Spain and France.

Whether as an on-premise or a cloud solution, DocuWare provides all the components needed for Enterprise Content Management. DocuWare's solutions are highly secure, scalable, flexible to integrate, and use future-proof technologies.

For more information on DocuWare, visit www.docuware.com.

Figure 16: Page 3 of the DocuWare proposal