

ENSE 470 - Milestone 6

Group Name: GLHF

Group Members:

- Vincent Chan
- Thiago De Melo
- Shuaihao Zhao
- Joe Emmanuel Samano

March 29, 2018

Summary

Executive Summary

Technology used:

- MySQL
- MySQL Workbench
- Java
- xhtml
- css
- Hibernate
- PrimeFaces Framework

ATDD test cases: All pass



Changes

Things that were added:

- In user page, we added a software history table for the user to able to track his/her requests.
- Added sorting feature
- Added loading feedback
- Give the user the ability to cancel a request
- Added a pending request feature to keep track of the request that needs to be addressed

Note: Terms used for Approver are Approve and Deny

Terms used for Analyst are Grant and Deny



Software History

Acronym ↕	Software Name ↕	Request Date ↕	Result Date	Status ▾	View
-	Alternative Sewage System	29/03/18 - 12:04	-	Pending	⌵
-	Care Manager	29/03/18 - 11:20	-	Approved	⌵
-	Ambulance Schedule System	29/03/18 - 05:54	-	Pending	⌵
-	Cisco WebEx	29/03/18 - 05:53	29/03/18 - 05:55	Denied	⌵
-	Cisco WebEx	29/03/18 - 05:53	29/03/18 - 05:56	Granted	⌵
-	Ambulance Schedule System	29/03/18 - 05:12	-	Pending	⌵
-	myeHealth Alberta	29/03/18 - 05:00	-	Pending	⌵
-	Cisco WebEx	29/03/18 - 04:59	29/03/18 - 05:10	Denied	⌵
-	Ambulance Schedule System	29/03/18 - 04:59	-	Pending	⌵
AT-AT	Abdomen Tissue and Analysis Tool	29/03/18 - 04:57	-	Pending	⌵

Request a software

Email: Department: Select a Software If you have any preference, choose an approver.

Description:

200 characters remaining.

Submit

Request Overview Information

Request ID: 13

Status: Pending

Request Date: 29/03/18 - 12:04

Software: Alternative Sewage System

Requester Information

Name: User

Email: hell.requester@gmail.com

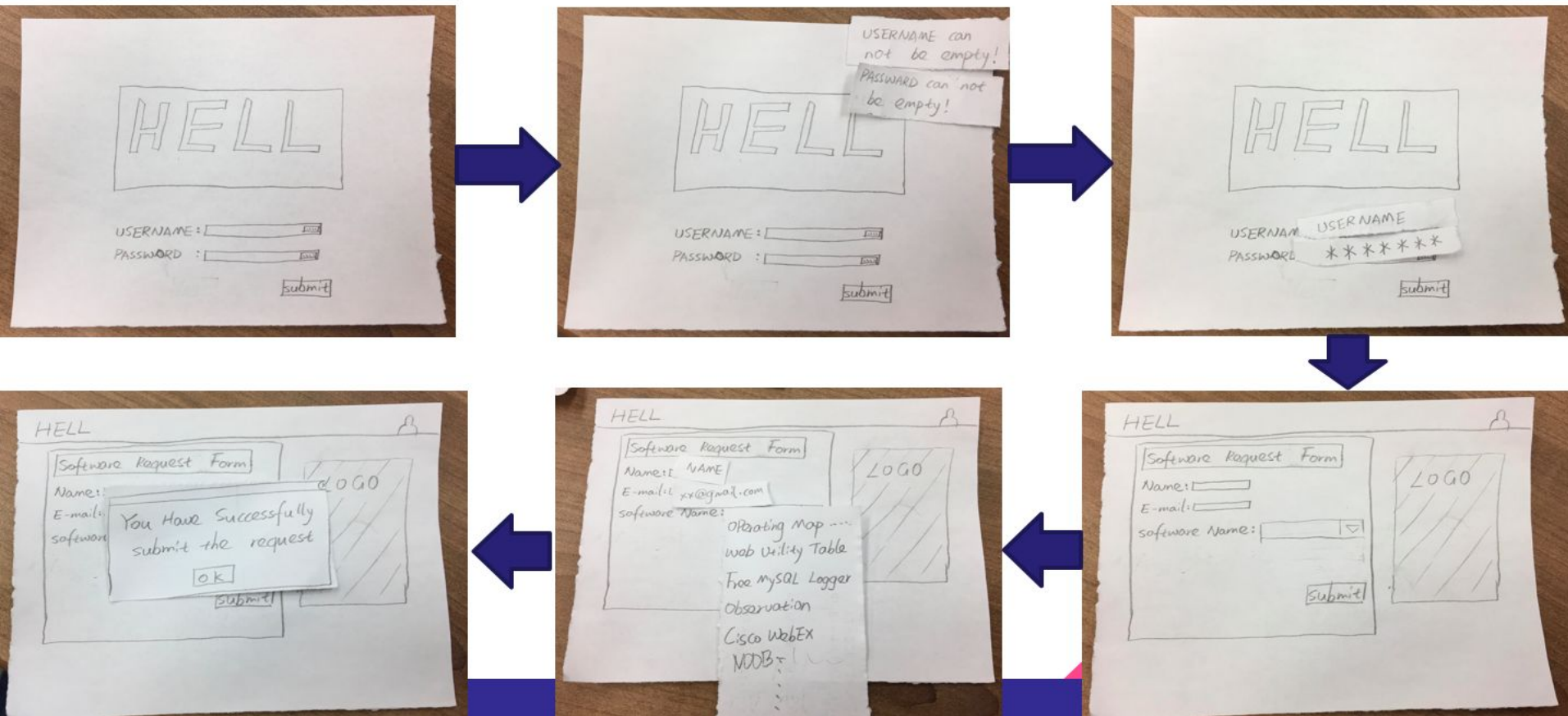
Cancel this request

You have 1 pending requests.

Submitting the request...



User



Approver

LOGO

[email](#)
[Index](#)

☐ HELL ----- 28/6/2008

HELL

USERNAME:

PASSWORD:

HELL

USERNAME

PASSWORD

HELL

client Name : D

e-mail : xxxx@gmail.com

software Request: Free MySQL Logger.
(FML)

State : Pending

HELL

clients Request List :

name	Software Request	State
A	Observation(version1)	Approved
B	Free MySQL Logger	Pending
C	Core Manager	Approved
D	Cisco WABEX	Pending
E	Homecore System	Approved
F	Pharma Core	Rejected

Analyst

LOGO

email

Index1

☐ HELL ----- 28/2/2008

HELL

USERNAME:

PASSWORD:

submit

HELL

clients Request List:

Name	Software Request	State	Approver
A	Observation	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	AA
B	Free MySQL Logger	Completed	BB
D	Cisco WebEx	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	CC
E	Homecare System	denied	DD

HELL

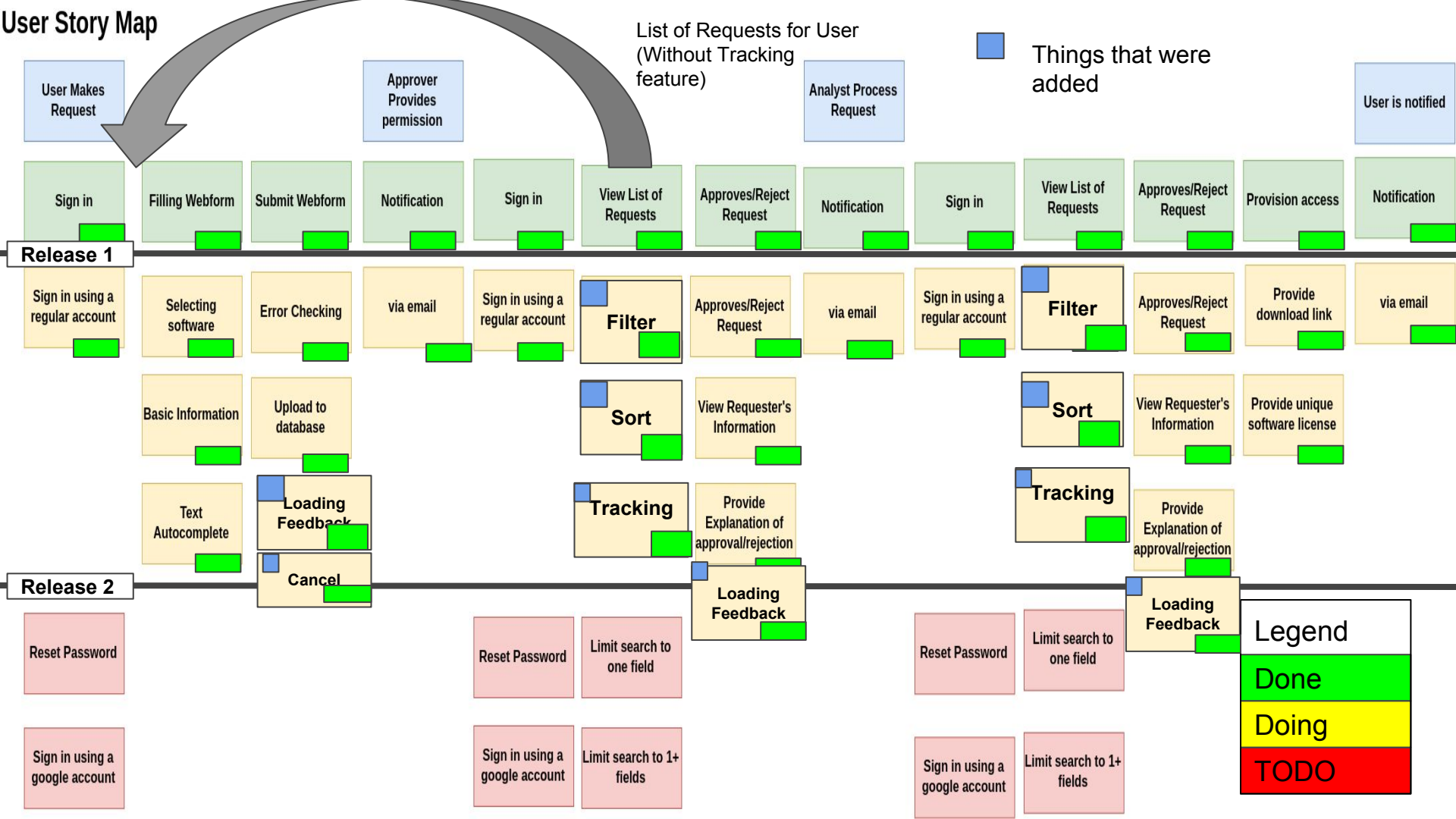
USERNAM

PASSWORD

submit

User Story Map

User Story Map



Demo

Link to website for our working tool:

[http://inteltec.com.br:8080/System/
pages/authentication.xhtml](http://inteltec.com.br:8080/System/pages/authentication.xhtml)

Username and Password

For user:

Username: User

All passwords:

Password: 12345

For approver:

Username: (First Name of
approver like Pete or Clay)

For analyst:

Username: Analyst



User focus: Software Requester/ Software Approver/ Software Analyst		
User story theme:		
As a user, I would like to have an account to sign in, so I can track my software requests easier.		
Acceptance test-driven development criteria (ATDD) (positive/negative)	Failed	Passed
[Positive] Given the user is currently on the login page, when s/he tries to log into his/her account with the correct password and username, then s/he will see the list of previous software requests on the left side of the screen and a webform on the right side of the screen.		*
[Negative] Given the user is currently on the login page, when s/he tries to log into his/her account with the incorrect password and username, then s/he will stay in the same page with a pop up indicating the message about the wrong operation.		*

User focus: Software Requester		
User story theme:		
As a user, I would like to have a main webpage showing my request so I can find the status of my previous requests		
Acceptance test-driven development criteria (ATDD) (positive/negative)	Failed	Passed
[Positive] Given the user is inside his account, when s/he have already submitted a request, then s/he will notice that there will exist a table that shows the status of these requests.		*
[Positive] Given the user is inside his account and is viewing the list of requests, when s/he types a value in the filter box, then s/he will see only a list of request that matches the filter.		*

User focus: Software Requester		
User story theme:		
As a user, I would like to make a software request to be able to use a particular software.		
Acceptance test-driven development criteria (ATDD) (positive/negative)	Failed	Passed
[Positive] Given that the user is logged in and is filling a webform, when s/he tries to search a particular software name by either typing in the name of the software or using the drop down structure, then s/he will notice that the software name suggestion that s/he can click on to appear in the web form.		*
[Negative] Given the user is logged in and is filling a webform, when s/he tries to search the particular software name by typing the incorrect software name, then s/he will not see a software name suggestion.		*
[Positive] Given the user is logged in and is filling a webform, when s/he has selected a particular software and searches for a particular approver, then s/he will notice the list of approvers for the particular software that s/he can choose.		*
[Positive] Given the user is logged in and is filling a webform, when s/he submits the webform, then the system will send the software approver an email to inform him about the new request.		*

User focus: Software Requester		
User story theme:		
As a user, I prefer filling the application form with error checking function, so I may not miss some important information.		
Acceptance test-driven development criteria (ATDD) (positive/negative)	Failed	Passed
[Negative] Given the user is filling the web form application, when s/he tries to submit the web form with all field empty, then s/he will notice a pop up on the right top of the page which contains the error message.		*
[Positive] Given the user is filling the web form application, when s/he tries to submit the web form by only fulfilling the software name, then s/he will notice that a new page will pop up with the success message and the software request information will appear in the table of the main page.		*
[Positive] Given the user is filling the web form application, when s/he tries to submit the web form by fulfilling all of the field, then s/he will notice that a new page will pop up with the success message and the software request information will appear in the table of the main page.		*

User focus: Software Analyst/Approver		
User story theme:		
As a software analyst/approver, I want to receive notifications via email so that I can easily monitor any software requests.		
Acceptance test-driven development criteria (ATDD) (positive/negative)	Failed	Passed
[Positive] Given the software analyst/approver is logged in to his/her email, when he/she reviews his/her inbox, then he/she will see an email showing that there is a new software request.		*
[Positive] Given the software analyst/approver has clicked on the email notification, when he/she reviews the content of the email, then he/she will see a link that will directly bring him/her to the sign in page.		*

User focus: Software Approver/Analyst		
User story theme:		
As a software approver/analyst, I would like to approve software request digitally to conveniently approve these software requests in a swift manner		
Acceptance test-driven development criteria (ATDD) (positive/negative)	Failed	Passed
[Positive] Given the software analyst/approver is signed in to his/her account, when he/she sees a list of requests and clicks on a request, then he/she will see a pop up showing the information about the request.		*
[Positive] Given the software approver/analyst is logged into the system and opened the prompt for a particular software request, when s/he select to either approve or deny the request, then s/he will see a confirmation message and then s/he will see a status change of the request.		*
[Positive] Given the software approver is logged into the system, when s/he changes the status of a software request, then the software will send an email to the IT analyst that the software request has been processed.		*
[Positive] Given the software analyst already chose to approve a request, when he/she confirms on the confirmation message, then an email notification will be sent to the requester containing the access to the software and the software analyst will be redirected to the page containing the list of software requests.		*
[Positive] Given the software analyst already chose to deny a request, when he/she confirms on the confirmation message, then an email notification will be sent to the requester a message showing declination of request and the software analyst will be redirected to the page containing the list of software requests.		*

Design Patterns

Observer

A way of notifying change to a number of classes.

We applied it where the system observes events like when a user press on the submit button, the system would send an email to the approver.



Model-view-controller

Model-view-controller (MVC) is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts.

An example of how we applied this architecture is with a web application where the user submits a request. In this case, the controller is the webform then the controller will manipulate the model which is the database and then the database will update the table in user's page which is the view part.



Factory

Creates an instance of several derived classes

An example of how we used this design pattern is where depending on the user we instantiate different sessions like for user we have a RequestBean class or for approver we have a ApproverDashboard class.




Chain of responsibility

A way of passing a request between a chain of objects

An example of how we applied this design pattern is we could say that the chain of objects are the user, approver, and analyst the request being passed between these objects is the user request.



Group Reflection

- How did you feel about this milestone? What did you like about it? What did you dislike?
 - Team members find their correct roles in the project in this milestone, and team members work together to complement their weaknesses and strengths
 - After couples of milestones effort, it's easier to do this milestone
 - It's hard to design the artistic website
 - What did you learn about yourself as you collaborated and worked through this milestone?
 - Reviewing the knowledge about the web programming and how it can be applied with java
 - Practicing the design patterns that we learned in class
 - How will you use what you have learned going forward?
 - We could apply the things we learned from building the project to future projects.
 - What “stuff & things” related to this milestone would you want help with?
 - Figuring out what kind of design patterns we used
 - We thought that we could have more time when doing our demo to we could go through all the features we added.
- 

Sources

Definitions for the design patterns

https://sourcemaking.com/design_patterns

<https://en.wikipedia.org/wiki/Model-view-controller>

