Project 3 kickoff

The project team

Hogeschool Rotterdam
Rotterdam, Netherlands

Introduction

Introduction

Overview

- Looking back
- A short overview of the project
- Survival tips

Looking back

Looking back

You are more than halfway through the first year.

Looking back

You are more than halfway through the first year.

The worst is over.

Looking back

You are more than halfway through the first year.

The worst is over.

You are on your way to becoming true professionals, and we could not be happier about it.

Looking back

What have you really learned?

- You can now program[a]
- You can work in teams to compose your skills

[a] for a small enough value of *programming*

Looking back

What have you really learned?

You can build simple software products that work locally and in isolation (standalone applications).

Looking back

What next?

Let's add a stepping stone to what you are learning.

A short overview of the project

A short overview of the project

We live in a world where data is completely pervasive.
Everything, everywhere, becomes data somehow.

A short overview of the project

We live in a world where data is completely pervasive. Everything, everywhere, becomes data somehow.

But what do we do with billions of numbers in a table? How do we make sense of it?

A short overview of the project

We live in a world where data is completely pervasive. Everything, everywhere, becomes data somehow.

But what do we do with billions of numbers in a table? How do we make sense of it?

We **make it alive**.

A short overview of the project

We live in a world where data is completely pervasive. Everything, everywhere, becomes data somehow.

But what do we do with billions of numbers in a table? How do we make sense of it?

We **make it alive**.

We **draw it**.

A short overview of the project

In this project, you will...

- ...build an application made up of multiple, separate components;
- ...gather, process, store, and visualise data on a GUI.

A short overview of the project

You will find data about the city of Rotterdam

A short overview of the project

You will find data about the city of Rotterdam

The choice of data will be based on interest and the possiblity to generate useful information from it.

A short overview of the project

The data will be studied and analysed into an ERD and a physical model

A short overview of the project

The data will be studied and analysed into an ERD and a physical model

The data will then be automatically processed and saved on a database[a]

[a] reflecting the documents above. Duh.

A short overview of the project

An application will connect to the database

A short overview of the project

An application will connect to the database

The connection may be anything you like: manual (hardcore), or mediated by an object-relational-mapping (ORM)[a]

[a] Entity framework + LINQ, hibernate, etc.

A short overview of the project

A series of queries will get data from the database into the application

A short overview of the project

A series of queries will get data from the database into the application

You will then process the data, and visualise the results

A short overview of the project

Visualisation will be the fundamental aspect of the work

A short overview of the project

Visualisation will be the fundamental aspect of the work

Visualisation[a] makes it possible to understand and make sense of the data

[a] If you put everything on a map, then it gets pretty, understandable, and yo get lots of points.

A short overview of the project

Visualisation will be the fundamental aspect of the work

Visualisation[a] makes it possible to understand and make sense of the data

[a] If you put everything on a map, then it gets pretty, understandable, and yo get lots of points.

Make sure to include enough information to make it clear how the visualisation is to be understood

Survival tips

Survival tips

Issues

Working together on a project is not always so simple!

Survival tips

Issues

- Setting up structures
- Writing reasonable code
- Collaborating via source control

Survival tips

Code is like violence: if it does not work, write some more?

Survival tips

Code is like violence: if it does not work, write some more?

Actually, this is not true!

Survival tips

Like any other complex activity, usage of brains is highly recommended

Survival tips

Like any other complex activity, usage of brains is highly recommended

This translates to *think before you act*

Survival tips

Thinking, in this case, refers to deliberately architecting everything

Survival tips

Thinking, in this case, refers to deliberately architecting everything

UML-like structures help: **not as a formal tool, but as a thinking aid**

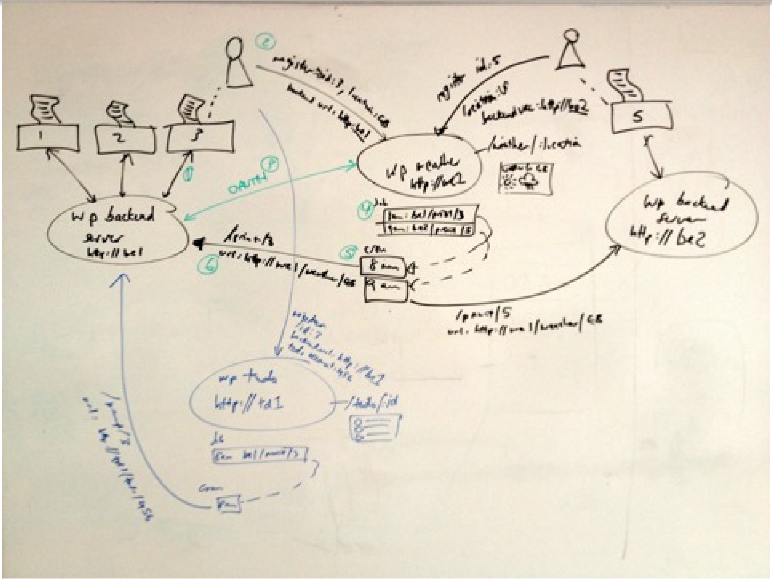
Survival tips

Thinking, in this case, refers to deliberately architecting everything

UML-like structures help: **not as a formal tool, but as a thinking aid**

It does not matter if your diagrams are perfect according to the standard: what matters is that they allow you to explore the domain of possible solutions.

HOGESCHOOL ROTTERDAM



Survival tips

Biggest goal: identify important blocks of functionality, and isolate them

Survival tips

Biggest goal: identify important blocks of functionality, and isolate them

Divide-and-conquer approach

Survival tips

Biggest goal: identify important blocks of functionality, and isolate them

Divide-and-conquer approach

The smallest blocks of functionality are classes

Survival tips

We are aware that the average student could potentially just write a single, big class doing everything

Survival tips

We are aware that the average student could potentially just write a single, big class doing everything

This would be disastrous for architecture, maintainability, and testing

Survival tips

We are aware that the average student could potentially just write a single, big class doing everything

This would be disastrous for architecture, maintainability, and testing

Worst of all: collaboration on such a monolithic entity becomes then impossible

Survival tips

We are aware that the average student could potentially just write a single, big class doing everything

This would be disastrous for architecture, maintainability, and testing

Worst of all: collaboration on such a monolithic entity becomes then impossible

You are required to write short classes that only cover one scenario

Survival tips

Small? Classes? Lol!

Complex functionality comes from composition and polymorphism

Survival tips

Life is too short to learn to use the command line

Survival tips

Life is too short to learn to use the command line

Use the GUI's: *GitHub desktop*, or the integration of GitHub in IDE's like Visual Studio and IntelliJ

Survival tips

Life is too short to merge commits

Survival tips

Life is too short to merge commits

Small files, and logical and short classes, already prevent a lot of conflicts

Survival tips

Life is too short to merge commits

Small files, and logical and short classes, already prevent a lot of conflicts

Ensure that each file has exactly **one and only one owner**

Survival tips

Commit often (once a day)

Survival tips

Commit often (once a day)

Waiting for days is a sure recipe for disaster

Survival tips

- Commit often (once a day)
- Waiting for days is a sure recipe for disaster
- Solve any commit conflicts right away, before going home

Conclusion

Conclusion

Closing up

- You have achieved a lot in the last months
- It is now time to impress us: build a modern data-heavy graphical application
- Be smart about your work, control complexity, and work together with discipline

This is it!

The best of luck, and thanks for the attention!