Lifecycle of HTTP request

RESTful API, resource(URI) and action

How Rails database work with database, object relational mapping

ORM (active record, data mapper)

Associations:

What is it for?

Write ruby code to create tables

Read schema.rb to create functions, generate getter and setter

MVC as company

Controller = boss

Guy doing work = model

Front desk = view

Rails framework

How does Rail pass parameters to application

Will be available in params

Diff between flash and session, relationship with cookie

Related to HWs (HW2)

Ruby:

Everything is an object

Method call to object

Not accessing instance variable, but making a function call

Every operation is a function call

Validate is also a function call

Ex. Login filter, checking workflow

When I am on… , and I do.. also a function call

Domain specific language

BDD and TDD

BDD, behavior driven

Meet the specification of behavior

With cucumber, write user story first, then step definition

TDD, test driven

Also tested in cucumber

Cucumber is a tool that symbolized both BDD and TDD

Development model

Agile and Waterfall

Waterfall:

Planning ahead

Development

Flow appear only once

Very detailed specification of everything

Spiral

More flow, more iterations

Agile

Extreme version of Spiral

Each iteration short and fasts

frequent meeting with customer

BDD and TDD

Don’t draw specific, but Lo-Fi

Understand HW

55 MC, one or multiple

Sinatra

Haml and erb, no detail in syntax, but concept

Sinatra lightweight framework, not MVC, page controller

MVC: Every model must have a controller, Sinatra not.

Receiver.send(operator, parameter)

ActiveRecord and DataMapper, difference

ActiveRecord

Every model knows how to save itself in database

Rb file inherite from Activerecord base

Use when creating new app to market, more flexible, easier to use

DataMapper

Knows nothing about database

Need help from other class, when try to save to database

Use in existing market

Schema

Used to prepare database for test