Assignment 2

Assignment structure

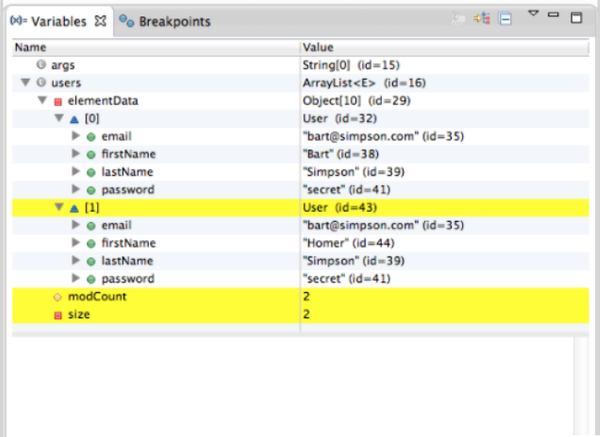
- 100% CA (2 Assignments)
  - **Assignment 1 (40%)** – due Sunday, 4th November @ 5PM (hard deadline).
    - No presentation required, but you may be asked to do a code walkthrough.
  - **Assignment 2 (60%)** – due Wed, 2nd January @ 9AM (hard deadline).
    - Specified in week 8 (just after midterm).
    - Presentation in early January required; a PPT template will be provided.

Agenda

Assignment 1
- Labs
- Grading
- Solution

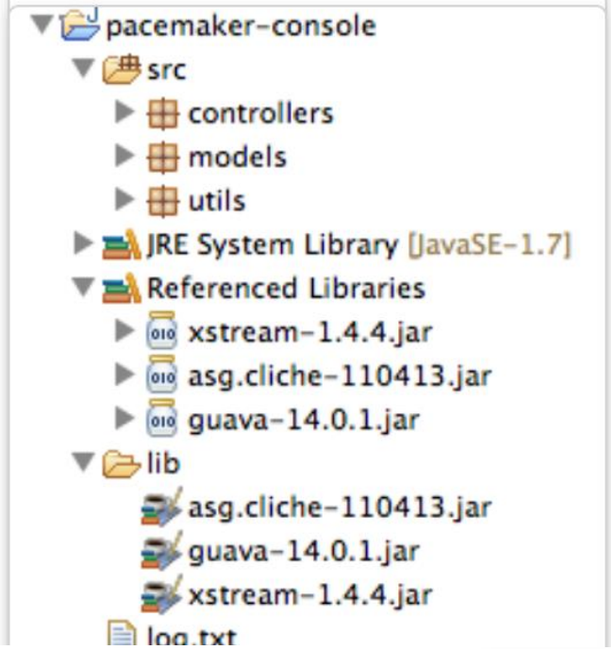
Assignment 2
- Briefing
- Grading
- Labs

Lab-01 Eclipse & Java

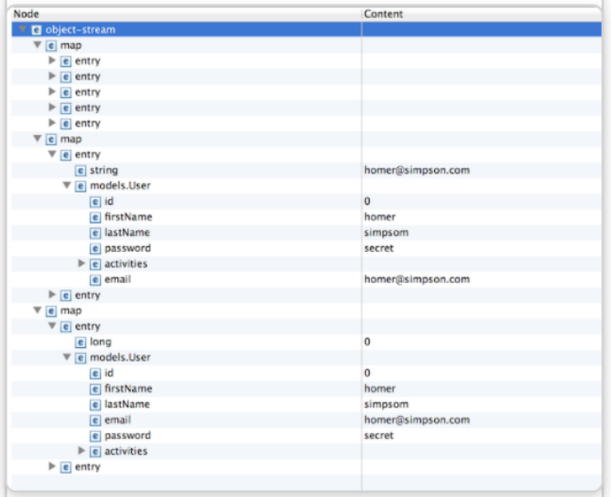


The objective of this lab is
and configure your devel
workstation and to bring i
some of the terms and c
covered in the opening l

Lab-02 CLI & Classes

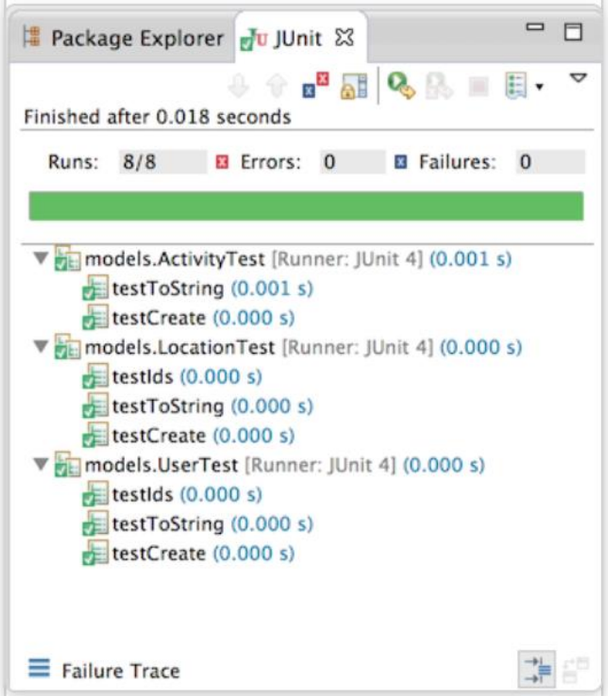


Lab-03 Objects & Serialization

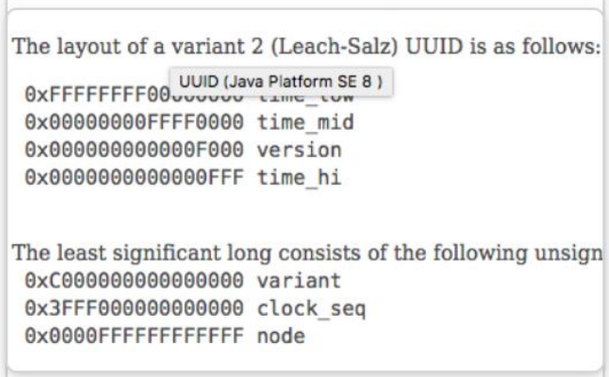


Extend the pacemaker application

Lab-04 Testing



Lab-05 Refactoring



Refactor pacemaker to employ uuid instead of long ids. Unsure the tests as still passing as we make this transition. Make a start command line formatting features.

Lab-06 Maven

In the previous lab, you installed Maven. In this lab, we will incorporate Maven into our pacemaker-console-lab05 solution. We will also use Maven to bring JUnit5 capabilities into Eclipse.

Standard	**Core Features [30%]**	**Presentation [20%]**	**Tests [30%]**	**Build Systems [20%]**
Baseline	Users/Activities/ Locations (lius, la, du)	Plain	basic API tests	none
Good	Start DateTime (la sortBy:) Persistence - XML (l, s)	Pretty	full API tests	maven (build)
Excellent	Persistence -JSON (cff)	Tabular	UI Tests	maven (test)
Outstanding	Persistence - YAML OR Extra Reports	Enhanced	accurate coverage report submitted	maven (modular approach)

Lab-08 Skeleton

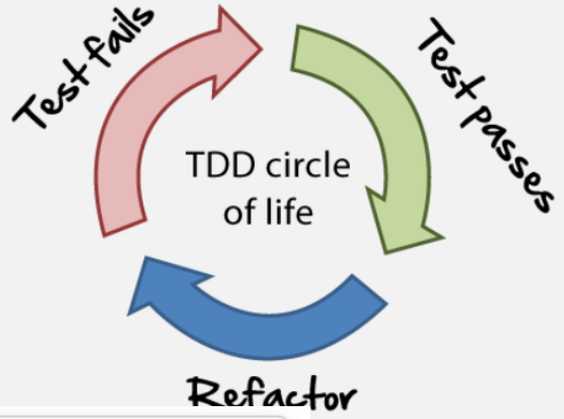
```
app.get("/users", ctx -> {
  service.listUsers(ctx);
});

app.post("/users", ctx -> {
  service.createUser(ctx);
});

app.get("/users/:id", ctx -> {
  service.listUser(ctx);
});

app.get("/users/:id/activities", ctx -> {
  service.getActivities(ctx);
});
```

Develop a baseline for Assignment 2, to include a simplified version of pacemaker application developed so far

Lab-09 Simple Rest API

Evolve a simple Rest
the existing pacemak
app using the J
mmicroframe

Lab-10 Rest CLI + Test

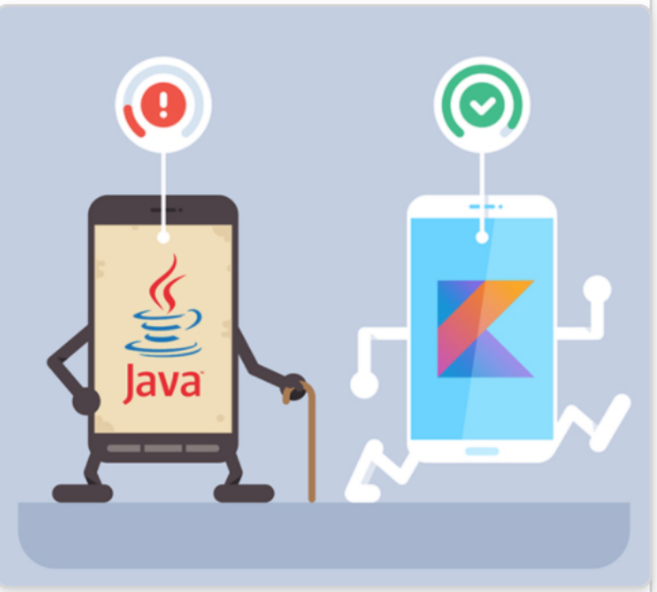


ent fo
d a mi
ervice.

Lab-11 Kotlin Rest Service

Rewrite aspects of the Pacemaker Skeleton Service in Kotlin. Verify that translation via the Java test and CLI clients.

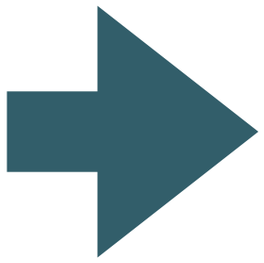
Lab-12 Kotlin Rest CLI + test



Rewrite the Java Test and CLI clients in Kotlin.

Assignment : Pacemaker 2.0

Create a new version of Pacemaker, evolved to explore 4 lines of inquiry



- Commands/Features
- Test Driven Development Practices
- Build & Deployment
- Language Features

Commands/Features (1/4)

List Users: List all users emails, first and last names	gu get-users ()
Register: Create an account for a new user	ru register-user (first name, last name, email, password)
Login: Log in a registered user in to pacemaker	lu login-user (email, password)
Logout: Logout current user	l logout ()
Add activity: create and add an activity for the logged in userr	aa add-activity (type, location, distance)
List Activities: List all activities for logged in user	la list-activities ()

Commands/Features (2/4)

Add location: Append location to an activity"	al add-location (activity-id, lat, lng)
List Activity Location: List all locations for a specific activity	lal list-activity-locations (activity-id)
ActivityReport: List all activities for logged in user, sorted alphabetically by type	ar activity-report ()
Follow Friend: Follow a specific friend	f follow (email)
List Friends: List all of the friends of the logged in user	lf list-friends ()
Friend Activity Report: List all activities of specific friend, sorted alphabetically by type	far friend-activity-report (email)

Commands/Features (3/4)

Activity Report: List all activities for logged in user by type. Sorted longest to shortest distance	ar activity-report (byType: type)
Unfollow Friends: Stop following a friend	uf unfollow-friend ()
Message Friend: send a message to a friend	mf message-friend (email, message)
List Messages: List all messages for the logged in user	lm list-messages ()
Distance Leader Board: list summary distances of all friends, sorted longest to shortest	dlb distance-leader-board ()
Friend Activity Report: List all activities of specific friend, sorted alphabetically by type	ar activity-report (byType: type)

Commands/Features (4/4)

Distance Leader Board: distance leader board refined by type	dlbbt distance-leader-board-by-type (byType: type)
Message All Friends: send a message to all friends"	maf message-all-friends (message)
Location Leader Board: list sorted summary distances of all friends in named location	llb location-leader-board (location)

Test Driven Development Practices

Coverage

30%

40%

50%

65%

80%

(with mocking)

Build & Deployment

Eclipse project archive
- pacemaker-console

---

github repo
- pacemaker-console

---

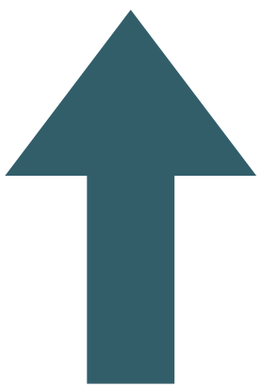
maven github repos:
- pacemaker-service
- pacemaker-console

---

pacemaker-service provides REST API
pacemaker-console access API (over http)

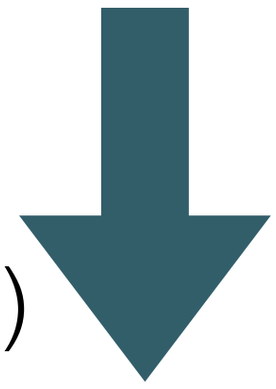
---

pacemaker-service deployed to cloud
pacemaker-client access cloud service



simple

useful



Language Features

verbose

Java

Java with Lambdas

Java with Streams OR Kotlin

Kotlin

concise

Grading Spectrum

Grade Band	Packaging & Deployment	Commands	TDD Coverage	Language
Starter	Eclipse project archive - pacemaker-console	**gu get-users ()** **ru register-user (first name, last name, email, password)** **lu login-user (email, password)** **l logout ()** **aa add-activity (type, location, distance)** **la list-activities ()**	30%	Java
Baseline	github repo - pacemaker-console	**al add-location (activity-id, longitude, latitude)** **lal list-activity-locations (activity-id)** **ar activity-report ()** **f follow (email)** **lf list-friends ()** **far friend-activity-report (email)**	40%	Java
Good	maven github repos: - pacemaker-service - pacemaker-console	**ar activity-report (byType: type)** **uf unfollow-friend ()** **mf message-friend (email, message)** **lm list-messages ()** **dlb distance-leader-board ()**	50%	Java with Lambdas
Excellent	pacemaker-service provides REST API pacemaker-console access API (over http)	**dlbbt distance-leader-board-by-type (byType: type)** **maf message-all-friends (message)** **llb location-leader-board (location)**	65%	Java with Streams OR Kotlin
Outstanding	pacemaker-service deployed to cloud pacemaker-client access cloud service	**Admin Account** **Define commands to administer service, to include:** **- remove users** **- disable/enable users** **- report user stats (nmr logins, average number of activities etc...)**	80% With Mocking	Kotlin

Lab 06

Lab-06 Maven

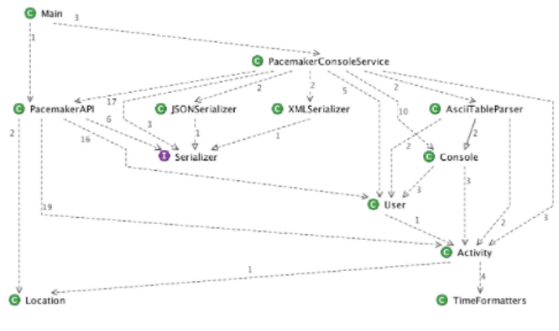
```
$ mvn -version
Apache Maven 3.1.0 (893ca28a1da9d5f51ac03827af98bb7
Maven home: /Users/edeleastar/dev/apache-maven-3.1.
Java version: 1.7.0_40, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk1.7
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.8.5", arch: "x86_
$
```

In the previous lab, you installed Maven. In this lab, we will incorporate Maven into our pacemaker-console-lab05 solution. We will also use Maven to bring JUnit5 capabilities into Eclipse.

Guidance on implementing maven modules

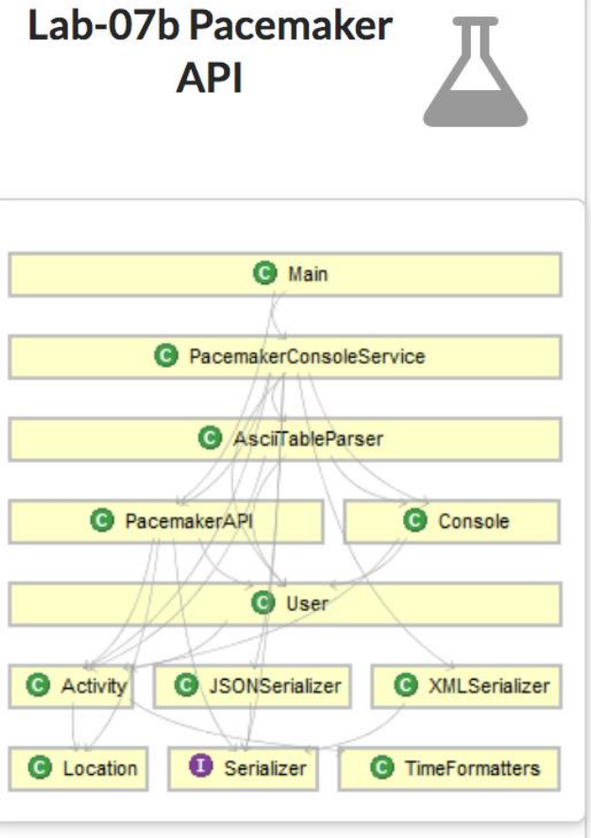
Packaging & Deployment
Eclipse project archive - pacemaker-console
github repo - pacemaker-console
maven github repos: - pacemaker-service - pacemaker-console
pacemaker-service provides REST API pacemaker-console access API (over http)

Lab-07a Pacemaker Models



Build a sample solu
Assignment 1, usin
techniques, Maven &

Lab-07b Pacemaker API



Complete the pacemaker assignment solution.

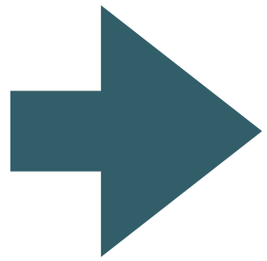
Guidance on overall project structure
Class Responsibilities
Code Formatting

Lab 08

Lab-08 Skeleton

```
app.get("/users", ctx -> {
  service.listUsers(ctx);
});

app.post("/users", ctx -> {
  service.createUser(ctx);
});

app.get("/users/:id", ctx -> {
  service.listUser(ctx);
});

app.get("/users/:id/activities", ctx -> {
  service.getActivities(ctx);
});
```

Develop a baseline for Assignment 2, to include a simplified version of pacemaker application developed so far



Commands
gu get-users () ru register-user (first name, last name, email, password) lu login-user (email, password) l logout () aa add-activity (type, location,

Provide foundation application structure + implement starter commands

Lab 09

Lab-09 Simple Rest API

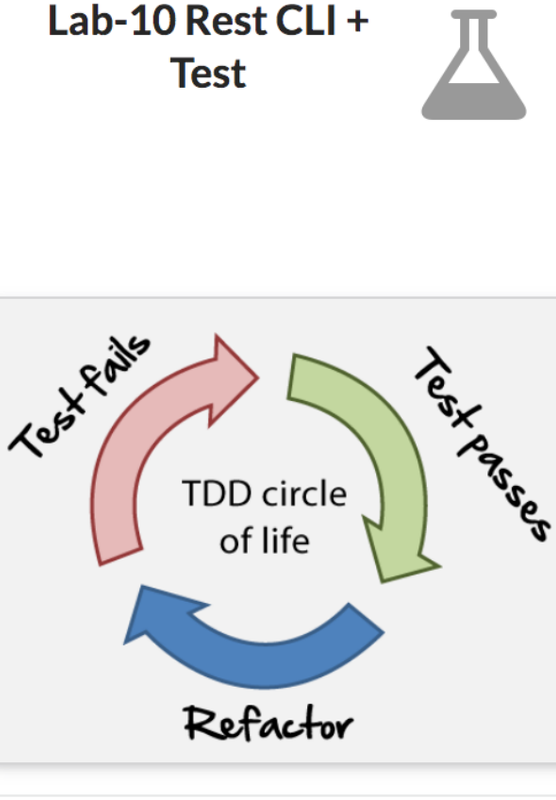
Evolve a simple Rest service from the existing pacemaker-skeleton app using the Javalin mmicroframework.

Evolve Starter commands into REST service

Packaging & Deployment
Eclipse project archive - pacemaker-console
github repo - pacemaker-console
maven github repos: - pacemaker-service - pacemaker-console
pacemaker-service provides REST API pacemaker-console access API (over http)

Lab 10

Lab-10 Rest CLI + Test



Compose a test client for the REST services. Also, build a minimal CLI for the service.

Evolve tests for APIs

TDD Coverage
30%
40%
50%
65%
80% With Mocking

Lab 11 & 12

Lab-11 Kotlin Rest Service

Kotlin

Rewrite aspects of the Pacemaker Skeleton Service in Kotlin. Verify that translation via the Java test and CLI clients.

Lab-12 Kotlin Rest CLI + test

Rewrite the Java Test and CLI clients in Kotlin.

Explore Kotlin implementations of pacemaker starter service

Language
Java
Java
Java with Lambdas
Java with Streams OR Kotlin
Kotlin