

T3am

SJSU - Summer 2018
CMPE 202

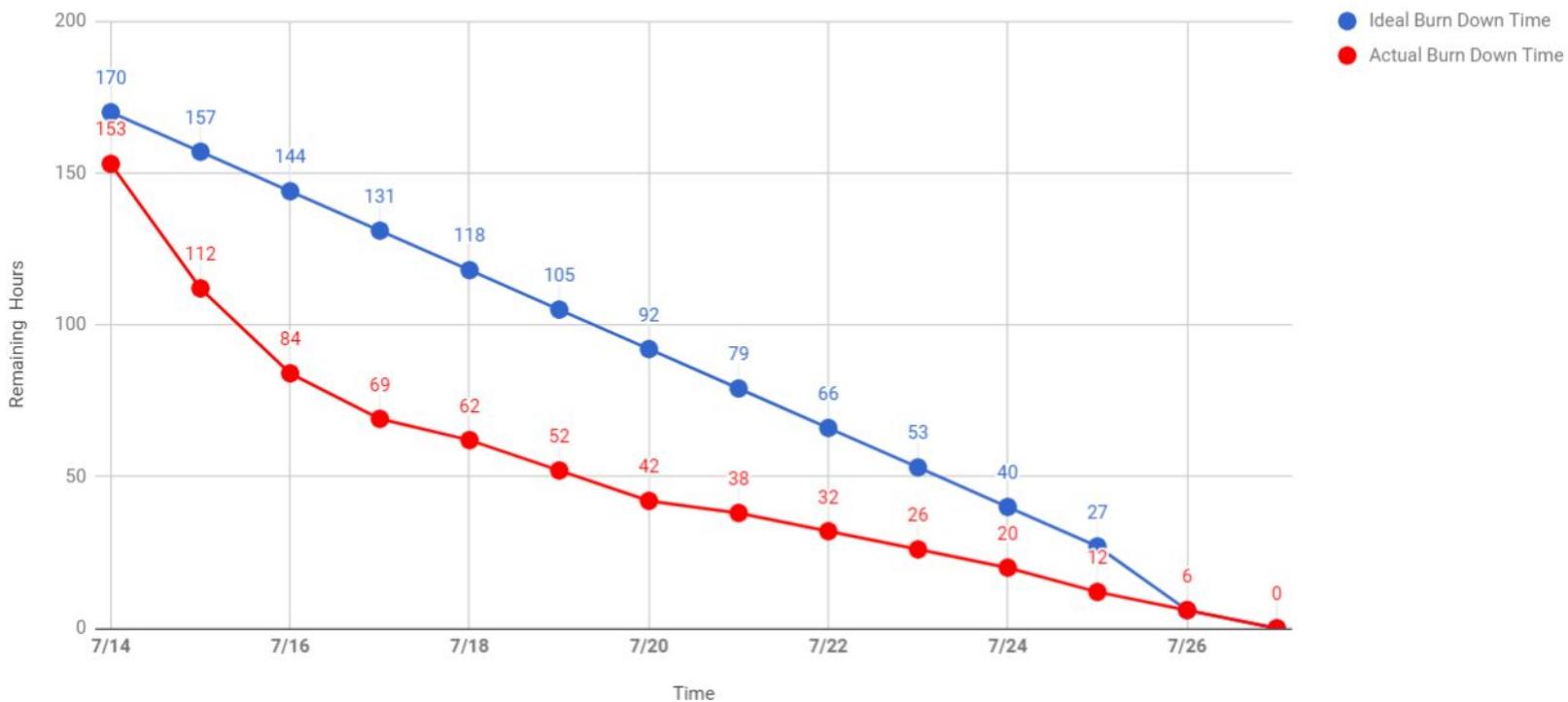
Professor Paul Nguyen
Professor Gopinath Vinodh

Members

- Sy Le (006088940)
- Hyunwook Shin (012507417)
- Kevin Lai (008498282)
- Lin Cheng (012484459)

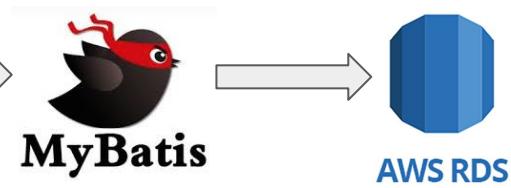
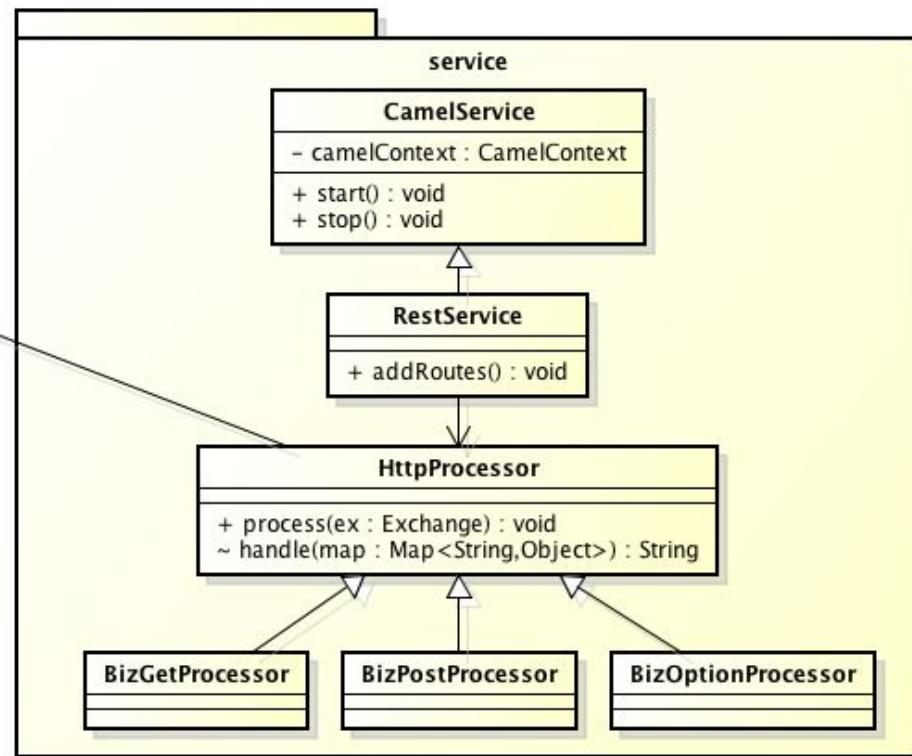
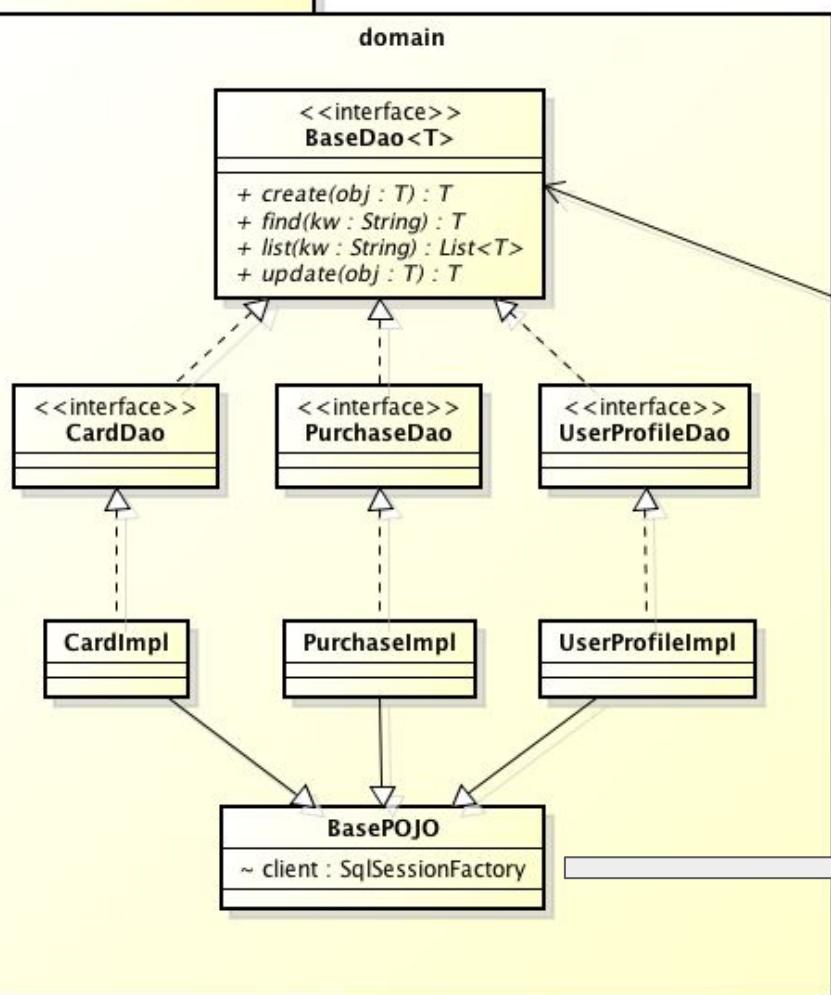
Team Sprint

Sprint Burndown Chart (Team T3am) (SJSU CMPE 202 Summer 2018)



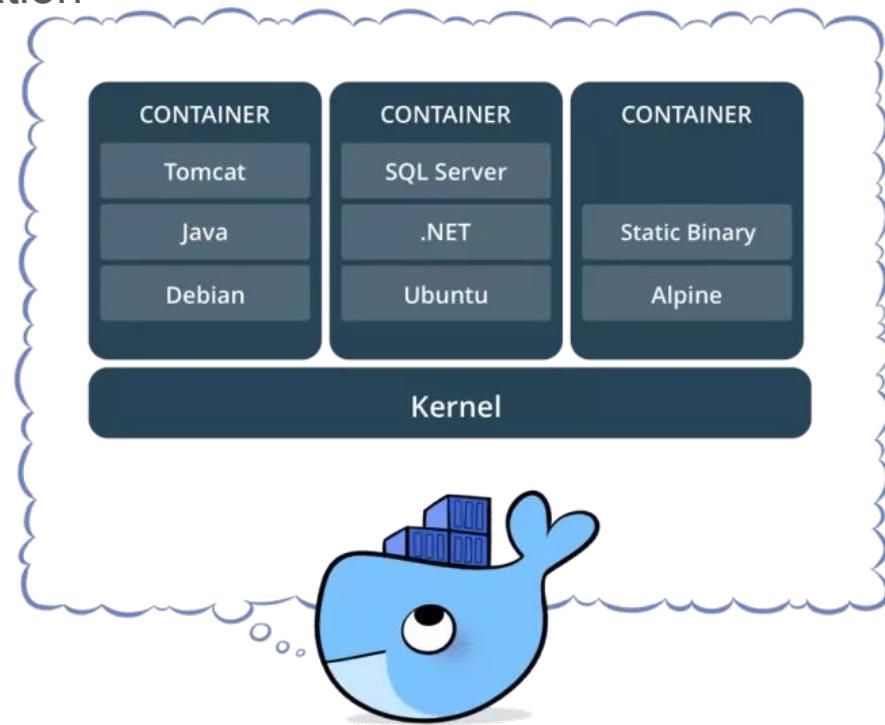
Back End

- API Server is based on Apache Camel
- Persistent data using MyBatis with AWS RDS
- Google Guice for Dependency Management
- Code Structure:
 - starbucks-common
 - Shared core components:
 - starbucks-domain
 - The persistent layer
 - DAO + MyBatis
 - starbucks-service
 - The API Server that wires up the app and serves up REST API's
 - Camel Restlet + Processors



Docker - Why Docker?

- Smaller than Virtual Machines
- Less bloatwares => Use Less Resource
- No more “Works on My Machine” situation
- Enhance DevOps and CI/CD



Docker - Step 1 - build the image

- docker build -t t3am
- The Dockerfile description

5 lines (4 sloc) | 109 Bytes

```
1 FROM java
2 EXPOSE 8202
3 COPY . /opt/t3am
4 CMD nohup java -cp /opt/t3am/starbucks2-service-1.0.jar RestService &
```

Docker - Step 2 - push the image upstream to AWS

- Push the docker image up to AWS Elastic Container Registry (ECR)
- docker tag t3am
914381644891.dkr.ecr.us-east-1.amazonaws.com/t3am
- docker push
914381644891.dkr.ecr.us-east-1.amazonaws.com/t3am

```
[root@ip-172-31-3-105 t3am]# docker tag t3am 914381644891.dkr.ecr.us-east-1.amazonaws.com/t3am
[root@ip-172-31-3-105 t3am]# docker push 914381644891.dkr.ecr.us-east-1.amazonaws.com/t3am
The push refers to a repository [914381644891.dkr.ecr.us-east-1.amazonaws.com/t3am]
a76d79f03ef5: Pushed
35c20f26d188: Layer already exists
c3fe59dd9556: Layer already exists
6ed1a81ba5b6: Layer already exists
a3483ce177ce: Layer already exists
ce6c8756685b: Layer already exists
30339f20ced0: Layer already exists
0eb22bfb707d: Layer already exists
a2ae92ffcd29: Layer already exists
latest: digest: sha256:816288955be83bf80de4d9cb02703024897909d8a1e944ad65cb65c8b3160969 size: 2212
```

Docker - Step 3 - Pull and Run Image from AWS

- docker pull
914381644891.dkr.ecr.us-east-1.amazonaws.com/t3am
- docker run -p 8202:8202 t3am

```
[root@ip-172-31-1-174 ~]# docker pull 914381644891.dkr.ecr.us-east-1.amazonaws.com/t3am
Using default tag: latest
latest: Pulling from t3am
e12c678537ae: Already exists
8d9ed335b7db: Already exists
3318dd58ae60: Already exists
624ba6156166: Already exists
c7a02d193df7: Already exists
813b62320378: Already exists
81cf5426393a: Already exists
0a2b7222259b: Already exists
fffff8146f0c[root@ip-172-31-1-174 ~]# docker run -p 8202:8202 t3am
Digest: sha256:log4j:WARN No appenders could be found for logger (org.apache.ibatis.logging.LogFactory).
Status: Downloading...
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
INFO [main] - Loaded 222 type converters
INFO [main] - Apache Camel 2.19.3 (CamelContext: camel-1) is starting
INFO [main] - JMX is enabled
INFO [main] - Runtime endpoint registry is in extended mode gathering usage statistics of all incoming and outgoing endpoints
INFO [main] - StreamCaching is not in use. If using streams then its recommended to enable stream caching. See more details at
INFO [main] - Route: route4 started and consuming from: direct://bizoption
INFO [main] - Route: route5 started and consuming from: direct://bizget
INFO [main] - Route: route6 started and consuming from: direct://bizpost
Jul 17, 2018 6:21:27 AM org.restlet.engine.connector.NetServerHelper start
INFO: Starting the internal [HTTP/1.1] server on port 8202
INFO [main] - Route: routel started and consuming from: http://localhost:8202/api/v1/(path)?restletMethods=OPTIONS
INFO [main] - Route: route2 started and consuming from: http://localhost:8202/api/v1/(path)?restletMethods=GET
INFO [main] - Route: route3 started and consuming from: http://localhost:8202/api/v1/(path)?restletMethods=POST
INFO [main] - Total 6 routes, of which 6 are started.
INFO [main] - Apache Camel 2.19.3 (CamelContext: camel-1) started in 0.925 seconds
```

Back End - Docker container deployment in AWS

Container Name	Image
t3am-c	914381644891.dkr.ecr.us-east-1.amazonaws.com/t3am

Details

Port Mappings

Host Port	Container Port	Protocol
8202	8202	tcp

Healthcheck

Command	ping localhost
Interval	100
Timeout	3
Start period	100
Retries	3

Cluster : t3am														
Get a detailed view of the resources on your cluster.														
				Status ACTIVE										
				Registered container instances 0										
				Pending tasks count 0 Fargate, 0 EC2										
				Running tasks count 1 Fargate, 0 EC2										
				Active service count 1 Fargate, 0 EC2										
				Draining service count 0 Fargate, 0 EC2										
Services Tasks ECS Instances Metrics Scheduled Tasks														
Create Update Delete														
<input type="text"/> Filter in this page Launch type ALL Service type ALL														
<table><thead><tr><th><input type="checkbox"/></th><th>Service Name</th><th>Status</th><th>Service ty...</th><th>Task Defin...</th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td>t3am-s</td><td>ACTIVE</td><td>REPLICA</td><td>t3am-t:1</td></tr></tbody></table>					<input type="checkbox"/>	Service Name	Status	Service ty...	Task Defin...	<input type="checkbox"/>	t3am-s	ACTIVE	REPLICA	t3am-t:1
<input type="checkbox"/>	Service Name	Status	Service ty...	Task Defin...										
<input type="checkbox"/>	t3am-s	ACTIVE	REPLICA	t3am-t:1										

DB Schema

public.sb_user

Columns

□ user_id	CHARACTER VARYING
□ email	CHARACTER VARYING
□ full_name	CHARACTER VARYING
□ password	CHARACTER VARYING
□ balance	NUMERIC
□ date_added	BIGINT

public.sb_card

Columns

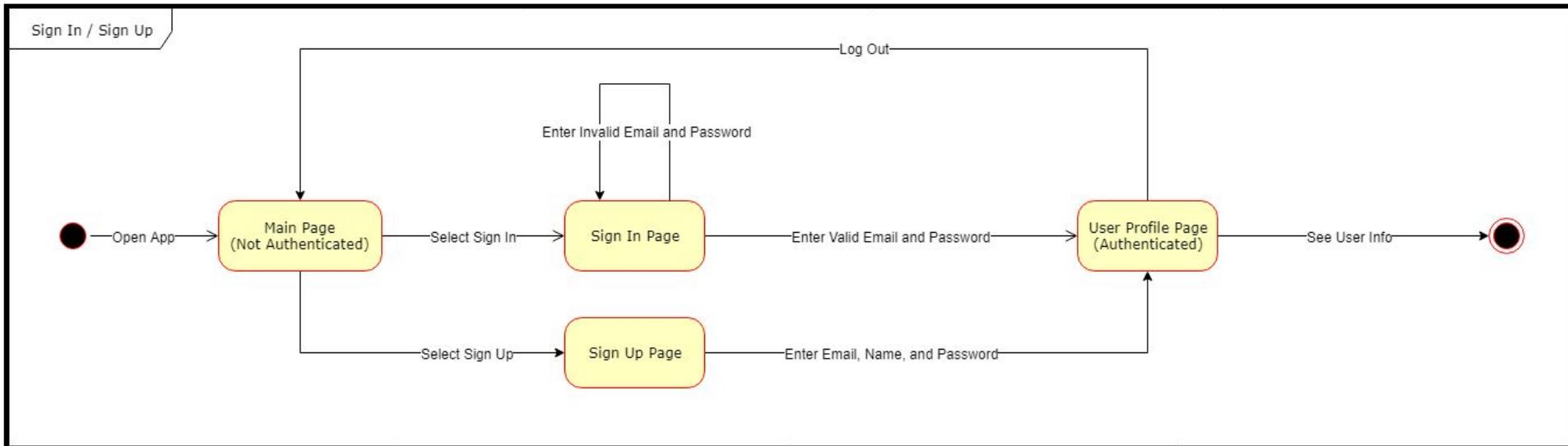
□ card_id	BIGINT
□ uid	CHARACTER VARYING
□ number	CHARACTER VARYING
□ code	CHARACTER VARYING
□ balance	NUMERIC
□ date_added	BIGINT

public.sb_purchase

Columns

□ purchase_id	BIGINT
□ uid	CHARACTER VARYING
□ balance	NUMERIC
□ date_added	BIGINT
□ note	CHARACTER VARYING

Front End - State Diagram



Front End - Technologies

- Bootstrap / jQuery UI
- jQuery
- Moment.js
- (live demo will be given at the end of the demo...)

Front End - Heroku Deployment

- App is deployed on Heroku

The screenshot shows the Heroku dashboard interface. At the top, there's a search bar labeled "Jump to Favorites, Apps, Pipelines, Spaces...". Below it, the user's profile icon and the word "HEROKU" are visible. In the center, there's a navigation bar with "Personal" and "cmpe202-t3am-starbucks-webapp". To the right of the app name are "Open app" and "More" buttons. Underneath, a GitHub integration section shows "synle/cmpe202-t3am-starbucks-webapp" with a "master" branch. A navigation menu below includes "Overview", "Resources", "Deploy", "Metrics", "Activity", "Access", and "Settings". The main content area displays "Free Dynos" and an "Upgrade to Hobby..." button. A message about "Web Dyno Autoscaling" is present. On the right, there's a "Hide" button and a "Learn More" link. At the bottom, a "web node index.js" entry is shown with a toggle switch, a price of "\$0.00", and a edit icon.

This screenshot shows a detailed view of the Heroku app configuration settings. It includes the following information:

Region	United States
Stack	heroku-16
Framework	Node.js
Slug Size	17.3 MiB of 500 MiB
GitHub Repo	synle/cmpe202-t3am-starbucks-webapp
Heroku Git URL	https://git.heroku.com/cmpe202-t3am-starbucks-webapp.git

Unit Testing and CI/CD

- JUnit for unit testing Java packages
- Python unit test for end-to-end testing
- Jenkins to deploy mock web-server and run end-to-end testing

```
python -m unittest api_tests
...
-----
Ran 3 tests in 0.249s
OK
dock2 ~/wses/su18-202-t3am/jenkins/tests$
```

```
python-requests/2.6.0 CPython/2.7.5 Linux/3.10.0-862.3.2.el7.x86_64 -
.
-----
Ran 3 tests in 0.635s
OK
make[2]: Leaving directory `/var/lib/jenkins/workspace/API Testing/su18-202-
t3am/jenkins/tests'
make[1]: Leaving directory `/var/lib/jenkins/workspace/API Testing/su18-202-t3am/jenkins'
+ set +x
sed -i "s/value=\"_your_db_password\"/value=\"****\"/g" starbucks2-
domain/src/main/java/sql-maps-config.xml
++ ps -ef
++ grep 'java.*jar'
++ grep -v grep
++ grep Rest
++ awk '{print $2}'
+ kill -9 3706
make: *** [run] Killed
Finished: SUCCESS
```

Team - Live Demo

- Demo Link
 - <http://cmpe202-t3am-starbucks-webapp.herokuapp.com>
 - username: **syle@gmail.com**
 - password: **password**