İstanbul
Bilgi Üniversitesi

LAUREATE INTERNATIONAL UNIVERSITIES

2018-2019 Education Year

Faculty of Engineering and Natural Sciences

Term Project

CMPE 331

Camply

https://camplycamping.herokuapp.com/

VOLKAN ARISLI

HANİFİ ENES GÜL

TUĞBERK GÖÇ

ARDA ÇAKMAKLI

ATAKAN KAYA

Project title
CAMPLY

Group members and group leader (project manager)
Volkan Arıslı-Coding

Arda Çakmaklı-Documentation

Hanifi Enes Gül-Analyzer

Tuğberk Göç-Tester
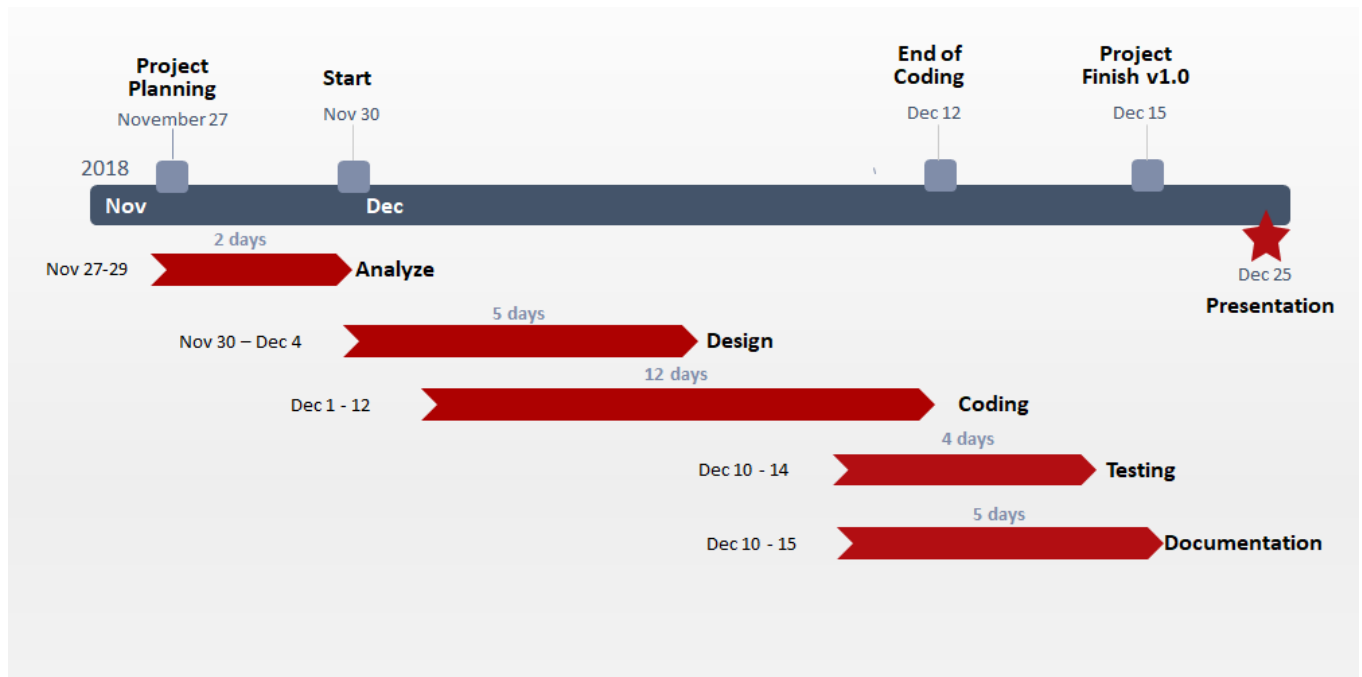
Atakan Kaya-Analyzer

Project description
Our aim is to create a platform that people who wants to camp can view available camp sites and can share their opinions with other fellow campers.
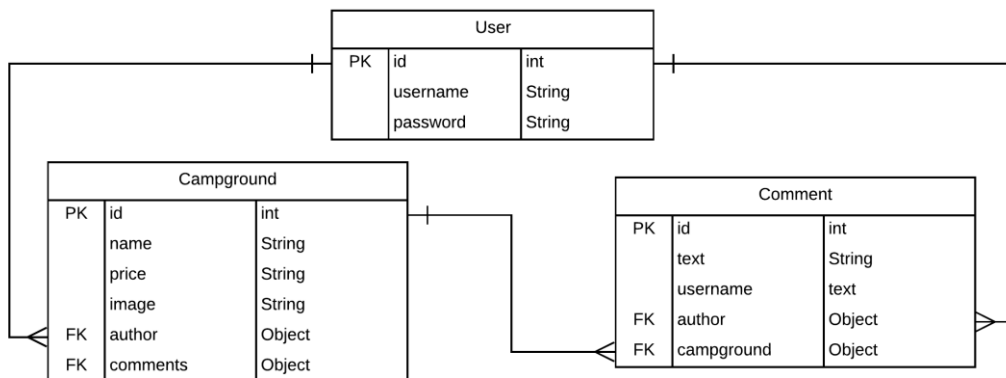
Project purpose
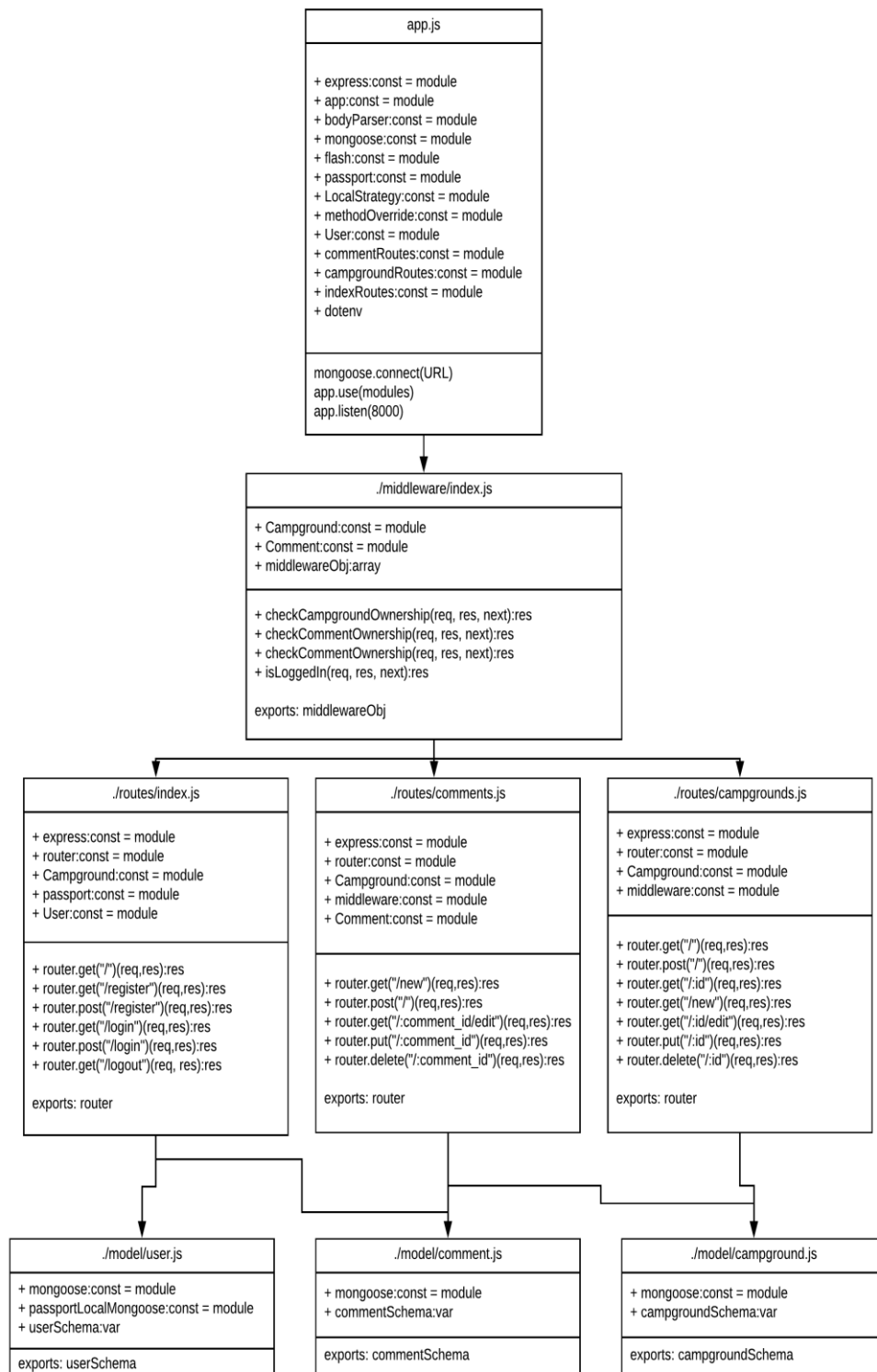Our goal is to not only make camping more popular and safe but also announce the new sites at the environment

Project planning and schedule

Structure of your Data Base

UML Diagram

app.js

+ express:const = module
+ app:const = module
+ bodyParser:const = module
+ mongoose:const = module
+ flash:const = module
+ passport:const = module
+ LocalStrategy:const = module
+ methodOverride:const = module
+ User:const = module
+ commentRoutes:const = module
+ campgroundRoutes:const = module
+ indexRoutes:const = module
+ dotenv

mongoose.connect(URL)
app.use(modules)
app.listen(8000)

./middleware/index.js

+ Campground:const = module
+ Comment:const = module
+ middlewareObj:array

+ checkCampgroundOwnership(req, res, next):res
+ checkCommentOwnership(req, res, next):res
+ checkCommentOwnership(req, res, next):res
+ isLoggedIn(req, res, next):res

exports: middlewareObj

./routes/index.js

+ express:const = module
+ router:const = module
+ Campground:const = module
+ passport:const = module
+ User:const = module

+ router.get("/")(req,res):res
+ router.get("/register")(req,res):res
+ router.post("/register")(req,res):res
+ router.get("/login")(req,res):res
+ router.post("/login")(req,res):res
+ router.get("/logout")(req, res):res

exports: router

./routes/comments.js

+ express:const = module
+ router:const = module
+ Campground:const = module
+ middleware:const = module
+ Comment:const = module

+ router.get("/new")(req,res):res
+ router.post("/")(req,res):res
+ router.get("/:comment_id/edit")(req,res):res
+ router.put("/:comment_id")(req,res):res
+ router.delete("/:comment_id")(req,res):res

exports: router

./routes/campgrounds.js

+ express:const = module
+ router:const = module
+ Campground:const = module
+ middleware:const = module

+ router.get("/")(req,res):res
+ router.post("/")(req,res):res
+ router.get("/:id")(req,res):res
+ router.get("/new")(req,res):res
+ router.get("/:id/edit")(req,res):res
+ router.put("/:id")(req,res):res
+ router.delete("/:id")(req,res):res

exports: router

./model/user.js

+ mongoose:const = module
+ passportLocalMongoose:const = module
+ userSchema:var

exports: userSchema

./model/comment.js

+ mongoose:const = module
+ commentSchema:var

exports: commentSchema

./model/campground.js

+ mongoose:const = module
+ campgroundSchema:var

exports: campgroundSchema

Software Process

We worked in a agile system which we showed in our planning schedule.We split the stages of the project in 5 pieces such as;Analyze,Design,Coding,Testing and Documentation.These stages started in different timelines because some of the stages are not connected.

Project testing

Our website's important stages are admins and users authorizations.These are the cases we tested in our project.

- Can admin edit and delete posts(expected:true/actual:true)
- Can admin edit and delete comments(expected:true/actual:true)
- Can none logged in user post camp sites. (expected:false/actual:false)
- Can none logged in user comment. (expected:false/actual:false)
- Can none logged in user sign up(expected:true/actual:true)
- Can logged in user post(expected:true/actual:true)
- Can logged in user comment. (expected:true/actual:true)
- Can logged in user edit their post and comment(expected:true/actual:true)
- Can logged in user edit and delete any post(expected:false/actual:false)
- Coverage test with using mocha library(expected:true/actual:true)

Libraries and Functions

- body-parser: "^1.18.3",
- connect-flash: "^0.1.1",
- dotenv: "^6.2.0",
- doxygen: "^0.3.2",
- ejs: "^2.6.1",
- env: "0.0.2",
- esdoc: "^1.1.0",
- esdoc-standard-plugin: "^1.0.0",
- express: "^4.16.3",
- express-session: "^1.15.6",
- method-override: "^3.0.0",
- moment: "^2.22.2",
- mongoose: "^5.2.9",
- nodemon: "^1.18.4",
- passport: "^0.4.0",
- passport-local: "^1.0.0",
- passport-local-mongoose: "^5.0.1",
- underscore: "^1.9.1"
- assert: "^1.4.1",
- checkCampgroundOwnership()//checks if logged in user owns that campground
- checkCommentOwnership()//checks if logged in own that comment
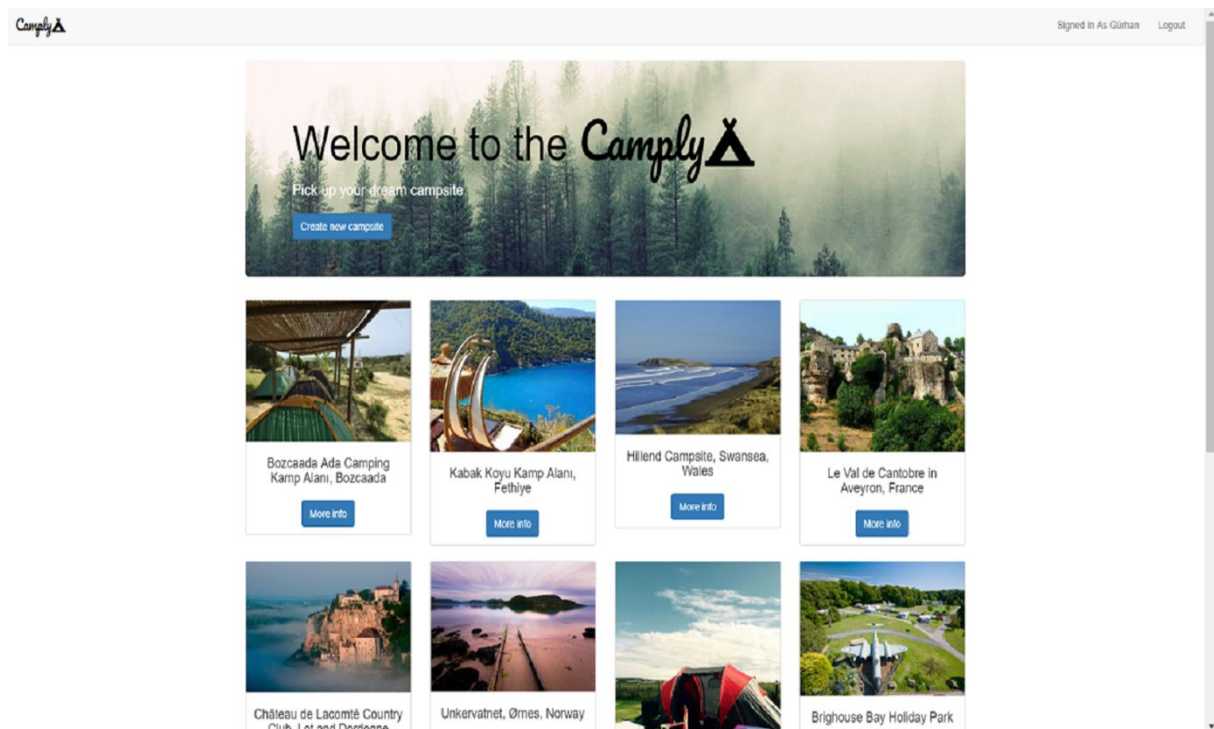- isLoggedIn()//checks if user logged in

Future Works

- Email authentication
- Ranking System
- User profiles
- Forum page
- Advising algorithm
- Campsite of the week
- False advertisement presentation algorithm
- Date of posts
- Upcoming events

Platforms used

- Visual Studio Code
- GitHub
- WebStorm
- Trello
- MongoLab
- Heroku

Screenshots

Homapage-Viewers can see all the campsites in the page

This is our campsite viewing page

This is our camp site creating interface

This is our user login page