Faham Khan, Kevin Mehta, Tyler Farkas
120976210, 130820360, 13067323

CP476 Internet Computer Project Phase - 2

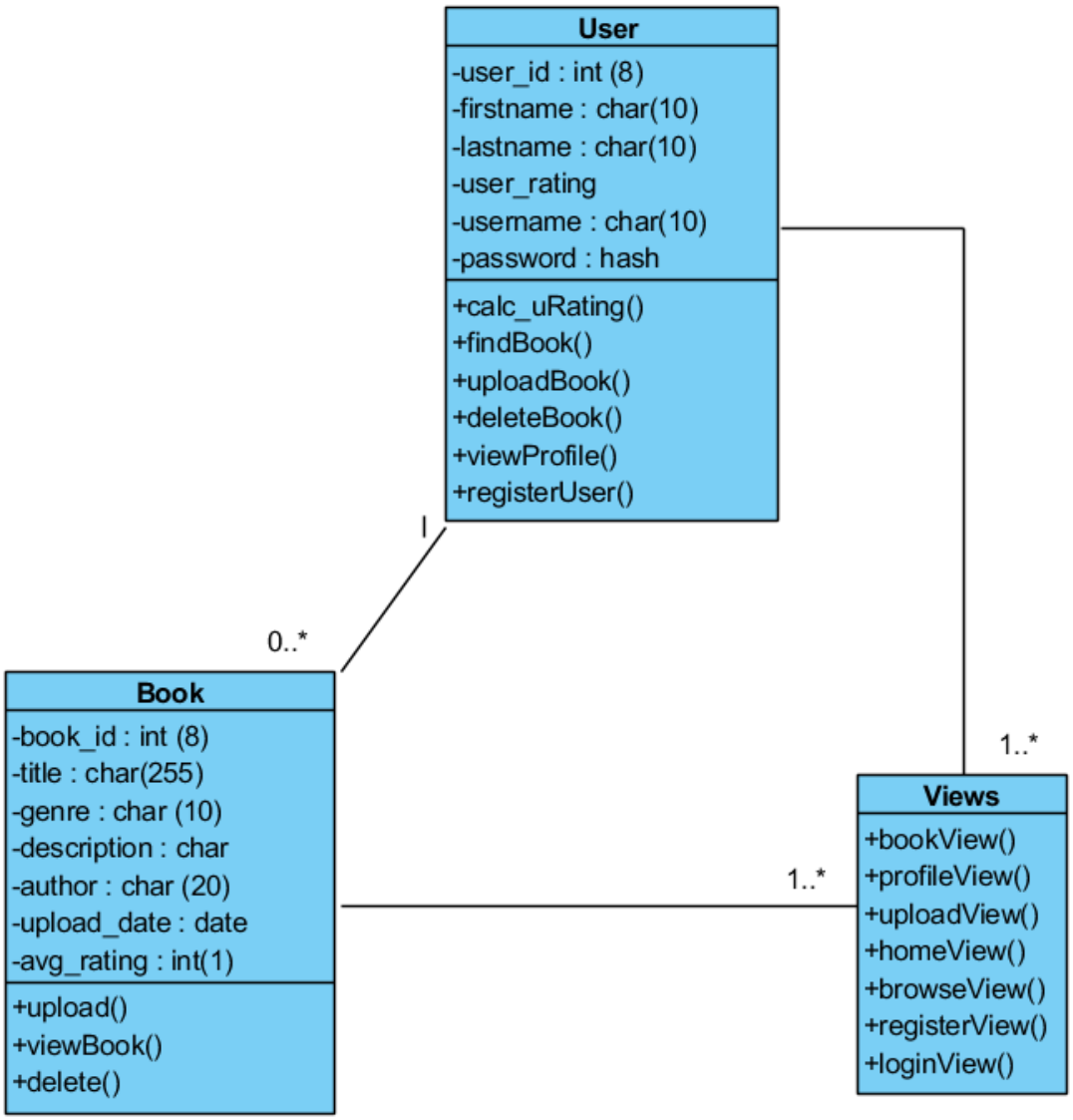
WRITERSBLOCK

Faham Khan, Kevin Mehta, Tyler Farkas
120976210, 130820360, 13067323

Table of Contents

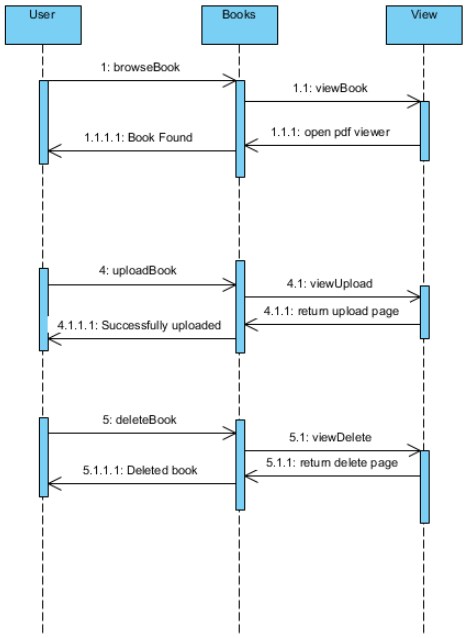
Faham Khan, Kevin Mehta, Tyler Farkas
120976210, 130820360, 13067323

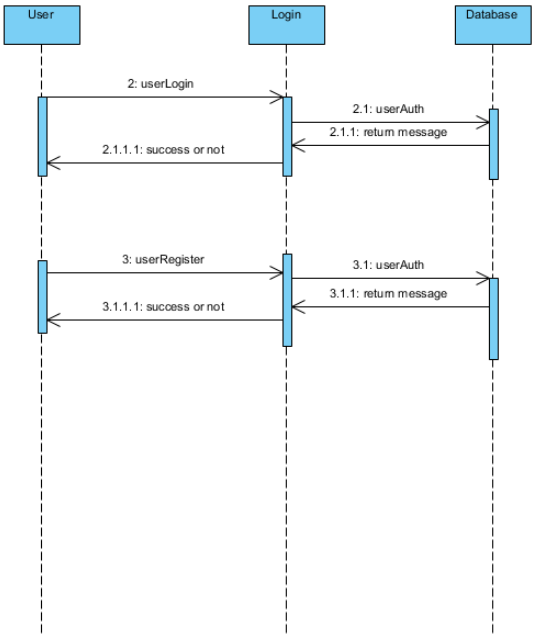
UML Class Diagram



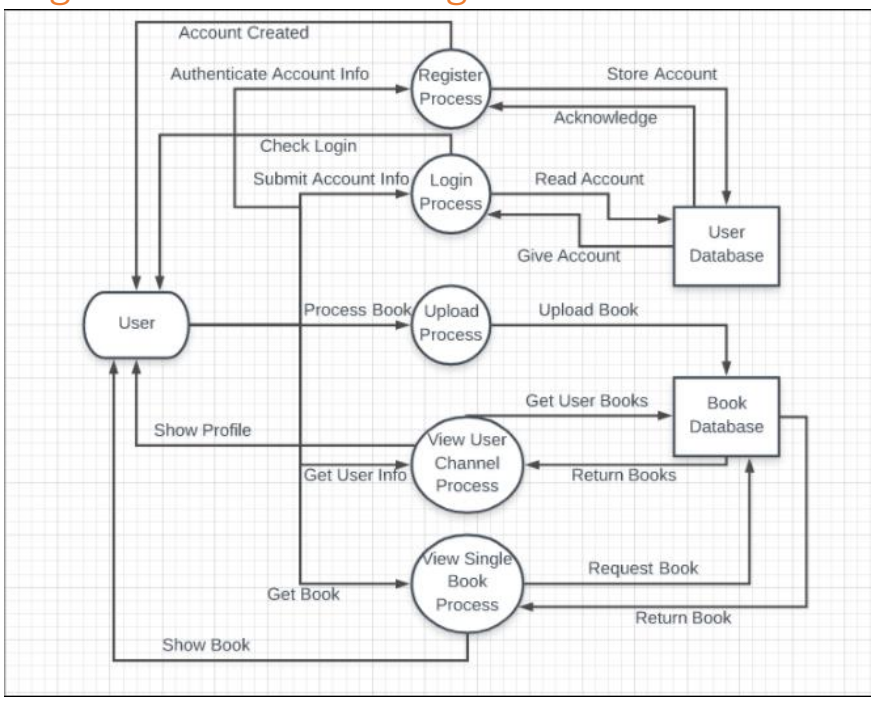
Faham Khan, Kevin Mehta, Tyler Farkas
120976210, 130820360, 13067323

UML Sequence Diagram





Logical Level 1 DFD Diagram



Entity Relationship Diagram

- N/A – We are using NoSQL which does not implement tables at all

Faham Khan, Kevin Mehta, Tyler Farkas
120976210, 130820360, 13067323

Application Interface Snippet

Code Snippets of classes, implementations, and clients that have been implemented…

User snippet

```
var express = require('express');
var router = express.Router();
var passport = require('passport');
var LocalStrategy = require('passport-local').Strategy;
var crypto = require('crypto');
var User = require('../models/user');
const uuidv1 = require('uuid/v1');


// Register
router.get('/register', function(req, res){
  if(req.isAuthenticated()){
    //Get username info
    res.redirect('/index')
  }
  else{
    res.render('register')
  }

});

// Login
router.get('/login', function(req, res){
  if(req.isAuthenticated()){
    //Get username info
    res.redirect('/index')
  }
  else{
    res.render('login')
  }

});
```

```
// Register User
router.post('/register', function(req, res){
  var name = req.body.name;
  console.log("THE NAME IS")
  console.log(name)
  var email = req.body.email;
  var username = req.body.username;
  var upassword = req.body.password;
  var upassword2 = req.body.password2;
  var Cloudant = require('@cloudant/cloudant');
  var me = '1f4f3453-d758-4393-a422-2010efd3d530-bluemix'; // Set this to your own account
  var cpassword = '28a1e839547250fe22c3e85e46c9f6200a26428ccd6fa95ade3778162b939779';
  var cloudant = Cloudant({account:me, password:cpassword});
  cloudant.db.list(function(err, allDbs) {
    console.log('All my databases: %s', allDbs.join(', '))
  });


   var newUser = {
    "name": name,
    "email": email,
    "username": username,
    "password": upassword,
    "book_ids" : "",
    "rating": 0,
    "book_count" : 0
    }
    console.log("Now creating the user")

   User.createUser(newUser, function(err, user){
     if(err) throw err;
     console.log("ESKITTIT"+newUser);
   })

   req.flash('success_msg', 'You are regist      n now login');
```

```
passport.use(new LocalStrategy(
  function(username, password, done) {
   User.getUserByUsername(username, function(err, user){
    if(err) throw err;
    if(!user){
      return done(null, false, {message: 'Unknown User'});
    }

    User.comparePassword(password, user.password, function(err, isMatch){
      if(err) throw err;
      if(isMatch){
        console.log("It is a match and the user id is id"+user._id)
        return done(null, user);
      } else {
        return done(null, false, {message: 'Invalid password'});
      }
    });
   });
  }));

passport.serializeUser(function(user, done) {
  console.log("And now onto serializeUser we have id of: "+user._id)
  done(null, user._id);
});

passport.deserializeUser(function(id, done) {
  User.getUserById(id, function(err, user) {
    console.log("Getting user by id: "+user)
    done(err, user);
  });
});
```

```
router.post('/login',
  passport.authenticate('local', {successRedirect:'/', failureRedirect:'/users/login',failureFlash: true}
  function(req, res) {
    req.flash('userinfo',req.user.username);

    res.redirect('/');
  });

router.get('/logout', function(req, res){
  req.logout();

  req.flash('success_msg', 'You are logged out');

  res.redirect('/users/login');
});


module.exports = router;
```

General code for the book

```
var express = require('express');
var router = express.Router();
var User = require('../models/user');
var Book = require('../models/book');
const fileUpload = require('express-fileupload');
var AWS = require('ibm-cos-sdk');
var util = require('util');
var fs = require('fs')
const uuidv1 = require('uuid/v1');

var config = {
    endpoint: 's3-api.dal-us-geo.objectstorage.softlayer.net',
    apiKeyId: 't4YOTrcghfoTuvquMx1Bkg2uchGEC9pJ6mxQb8LisCO5',
    ibmAuthEndpoint: 'https://iam.ng.bluemix.net/oidc/token',
    serviceInstanceId: 'crn:v1:bluemix:public:cloud-object-storage:glob
};

var cos = new AWS.S3(config);
```

```
router.get('/', function(req, res){
  console.log("The username here is "+req.user)
  var isLoggedIn = false
  var menuBar;
  if(req.isAuthenticated()){
    //Get username info
    User.getUserInfoById(req.user, function(err, user) {
      console.log("Getting user by id: "+user.username)
      var username = user.username
      res.render('index',
      { userID : username }
      )
    });

  }
  else{
    res.render('index',
    { userID : false }
    )
  }

});

function ensureAuthenticated(req, res, next){
  if(req.isAuthenticated()){

    return next();
  } else {
    console.log("Yea we are redirecting man")
    //req.flash('error_msg','You are not logged in');
    res.redirect('/users/login');
  }
}
```

```
router.get('/profile', ensureAuthenticated, function(req, res){

  res.render('profile')
});

router.get('/upload', function(req, res){
  res.render('upload')
});
```

```
router.post('/upload', function(req, res) {
  if (!req.files)
    return res.status(400).send('No files were uploaded.');

  // The name of the input field (i.e. "sampleFile") is used to retrieve the uploaded file
  let sampleFile = req.files.userfile;

  var id = uuidv1();
  var newBook = {
   "_id": id,
   "title": req.files.userfile.name,
   "genre": "",
   "description": "",
   "views": 0,
   "rating": 0,
   "authorID": req.user
   }
   Book.createBook(newBook, function(err, user){
     if(err) throw err;
     console.log("ESKITTIT"+newBook);
   })

  //var test= fs.createReadStream('public/pdfs/Portfolio-Culminating Assignment Part 2.pdf')
  //uploadFile(req.files.userfile);
  console.log()

  // Use the mv() method to place the file somewhere on your server
    sampleFile.mv('public/pdfs/'+id+'.pdf', function(err) {
    if (err)
      return res.status(500).send(err);

    res.send('File uploaded!');
  });
});
```

```
function uploadFile(info){
  console.log('Creating object');
  return cos.putObject({
          Bucket: 'books',
          Key: info.name,
          Body: fs.readFileSync('public/pdfs/'+info.name),
          ContentType: 'application/pdf'
  }).promise();
}


module.exports = router;
```

Faham Khan, Kevin Mehta, Tyler Farkas
120976210, 130820360, 13067323

Book class

```
var bcrypt = require('bcryptjs');
var Cloudant = require('@cloudant/cloudant');
var me = '1f4f3453-d758-4393-a422-2010efd3d530-bluemix'; // Set this to your own account
var password = '28a1e839547250fe22c3e85e46c9f6200a26428ccd6fa95ade3778162b939779';
var cloudant = Cloudant({account:me, password:password});
var searchBook = cloudant.db.use('books')

var Book = module.exports

module.exports.createBook = function(newBook, callback){
  Cloudant({account:me, password:password}, function(er, cloudant) {
    if (er)
      return console.log('Error connecting to Cloudant account %s: %s', me, er.message)

        // specify the database we are going to use
       // and insert a document in it
       searchBook.insert(newBook, function(err, body, header) {
         if (err)
           return console.log('[book.insert] ', err.message)

         console.log('you have inserted the book info.')
         console.log(body)
       })
  })
}
```

```
module.exports.getBookByBookname = function(bookname, callback){
  searchBook.search('book', 'newSearch', {q:'bookname:'+bookname}, function(er, result) {
    if (er) {
      throw er;
    }
    console.log("Now getting book")
    if(result.rows.length>0){
      searchBook.get(result.rows[0].id, { revs_info: true }, function(err, data) {
        console.log(`Document contents:` + JSON.stringify(data));
        console.log("And the id is "+data._id)
        callback(null, data)
      });
    }
    else{
      callback(null,false)
    }
  });
}

module.exports.getBookById = function(id, callback){
  searchBook.get(id, { revs_info: true }, function(err, data) {
    if(err){
      callback(null,false)
    }
    else{
      console.log("We successfully searched and here is the id: "+data._id);
      console.log(`Document contents:` + JSON.stringify(data));
      callback(null, data._id)
    }
  });

}
```

```
module.exports.updateUserInfo = function(user, callback){
  searchuser.get(user._id, { revs_info: true }, function(err, data) {
    if(err){
      callback(null,false)
    }
    else{
      console.log("We successfully searched and here is the id: "+data._id);
      console.log(`Document contents:` + JSON.stringify(data));

      Cloudant({account:me, password:password}, function(er, cloudant) {
        if (er)
          return console.log('Error connecting to Cloudant account %s: %s', me, er.message)

            // specify the database we are going to use
          var users = cloudant.db.use('users')
          // and insert a document in it
          users.insert(newUser, function(err, body, header) {
            if (err)
              return console.log('[users.insert] ', err.message)

            console.log('you have inserted the user info.')
            console.log(body)
          })
      })
      callback(null, data)
    }
  });
}
```

```
module.exports.comparePassword = function(candidatePassword, hash, callback){
  bcrypt.compare(candidatePassword, hash, function(err, isMatch) {
      if(err) throw err;
      console.log("And the match is "+isMatch)
      callback(null, isMatch);
  });
}


module.exports.getUserInfoById = function(id, callback){
  searchuser.get(id, { revs_info: true }, function(err, data) {
    if(err){
      callback(null,false)
    }
    else{
      console.log("We successfully searched and here is the id: "+data._id);
      console.log(`Document contents:` + JSON.stringify(data));
      callback(null, data)
    }
  });
}
```

Faham Khan, Kevin Mehta, Tyler Farkas
120976210, 130820360, 13067323

Login Form Snippet

```
section#signUp.bg-primary
  form(method='post', action='/users/login')
    .container
      h1 Login
      hr
      label Username
      input.form-control(type='text', name='username', placeholder='Username')
      label Password
      input.form-control(type='password', name='password', placeholder='Password')
      label
        input(type='checkbox', checked='checked', name='remember', style='margin-bottom:15px')
        |  Remember me
      .clearfix
        button.signupbtn2(type='submit') Login
```

Register Snippet

```
section#signUp.bg-primary
  form(method='post', action='/users/register')
    .container
      h1 Sign Up
      p Please fill in this form to create an account.
      hr
      label(for='username')
        b Username
      input(type='text', placeholder='Enter Username', name='username', required='')
      label(for='fullname')
        b Full Name
      input(type='text', placeholder='Enter Full Name', name='name', required='')
      label(for='email')
        b Email
      input(type='text', placeholder='Enter Email', name='email', required='')
      label(for='psw')
        b Password
      input(type='password', placeholder='Password', name='password', required='')
      label(for='psw-repeat')
        b Re-Enter Password
      input(type='password', placeholder='Password', name='password2', required='')
      label
        input(type='checkbox', checked='checked', name='remember', style='margin-bottom:15px')
        |  Remember me
      .clearfix
        button.cancelbtn(type='button') Cancel
        button.signupbtn(type='submit') Sig
```