

软件设计文档

BUGKILLER-GROUP



软件设计文档 (SDS)

本次小组实现电影购票系统网站，想要完成用户进行登录，自由顺序选择电影和电影院，完成对于电影和电影院还有个人信息的记录并购票过程

进行网站系统设计如下：

一， 需求分析

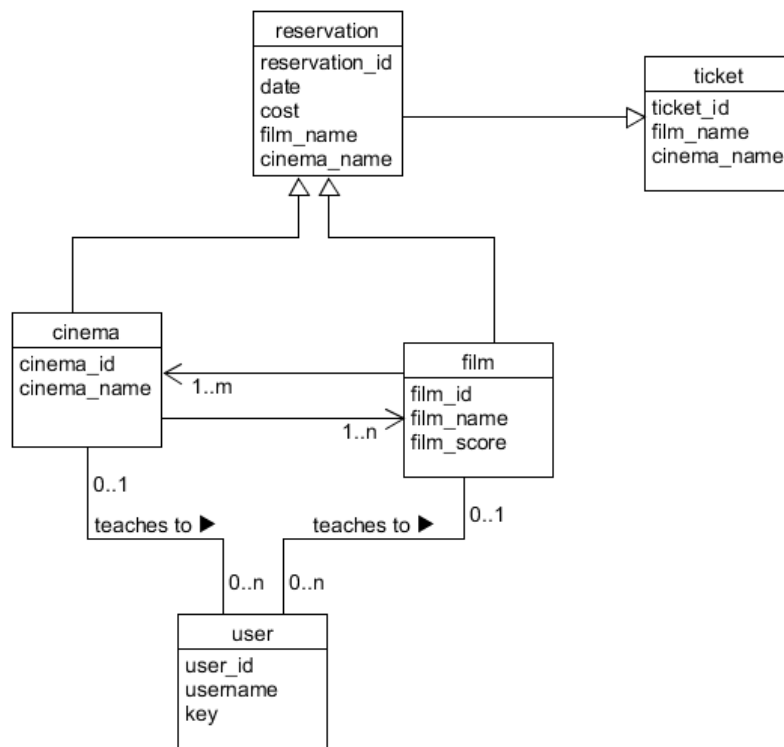
本次小组实现电影购票系统网站，

1. 能够完成用户的登录功能，提供 login.html 页面，在 index.html 页面有索引。（记录包括用户名称和密码）
2. 完成电影主页面电影推荐
3. 完成主页面下面的其他外网链接
4. 完成电影院和电影选择其中之一可以自行跳转到另一个的主页面留来下一步的选择，并记录信息
5. 完成提交后，能够显示用户名称，所选的电影院名称以及电影名称。
6. 用户确定后能够点击确定跳转到 succss 显示页面，并支持跳回主页面。
7. 每个页面要有各个可跳转页面的主控条，便于各个页面之间切换

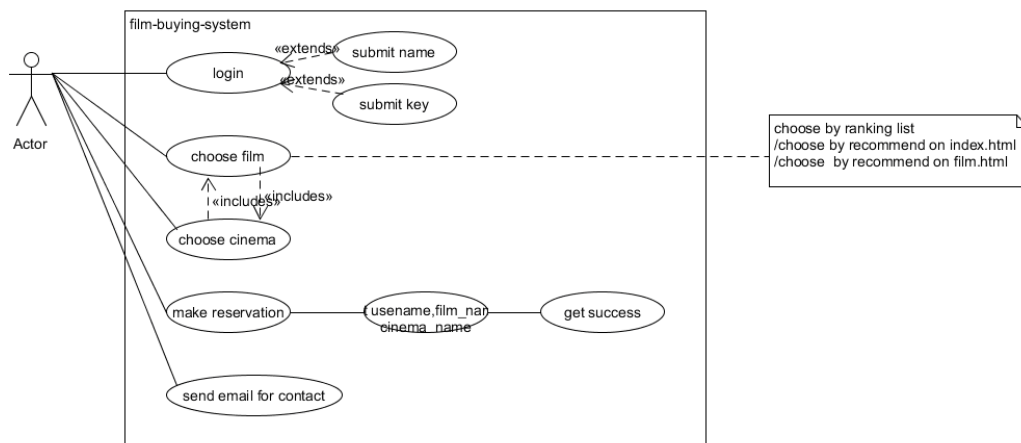
二， 模块概要设计

主要分为电影部分，电影院部分，数据部分（后端数据库以及 server 的调用和改写），用户信息部分。

进行林谷模型设计主要如下：



进行 use-case 设计如下所示：



三．系统架构说明：

前端主要使用 bootstrap,html5,css3 进行开发，
Server 使用 node.js

四. 功能设计

1. 主要实现用户的主三种选购方式
2. 能够通过服务器对于前端信息进行更改。
3. 记录用户的选择信息并最终展示给用户，对于用户易操作，且友好

化。

在主页面底端添加锚，使得快速回到顶端。

每个页面都有相同的侧边栏，选择某一页面会将侧边栏的相应图标字体变化显示。

对于页面结构进行规整，底端分为四个基本信息模块，每个页面顶端都有登录功能，便于用户任何时候登录。

nodejs 服务器与 javascript（位于 app.js 和.ejs 文件中）

涉及的技术与其描述	对应的代码文件中的具体位置
express 模块下, nodejs 基本功能（页面跳转）的实现：通过对 8081 端口的监听, 从 URL 获取客户想访问的页面, 通过 res.sendFile 函数把对应的 html 页面返回给前端。	<pre>app.get('/', function (req, res) { res.send('Hello World'); }) app.get('/index.html', function (req, res) { res.sendFile(__dirname + "/" + "index.html"); }) app.get('/film.html', function (req, res) { res.sendFile(__dirname + "/" + "film.html"); }) app.get('/cinema.html', function (req, res) { res.sendFile(__dirname + "/" + "cinema.html"); }) app.get('/login.html', function (req, res) { res.sendFile(__dirname + "/" + "login.html"); })</pre>
客户登录	<pre>app.post('/user_login', function(req, res) { myid = req.body.user_id; myname = req.body.user_name; // mytel = req.body.tel; // pic_id = req.body. console.log("the infos are: %s", myname); res.render('continue.ejs', { user_name: myname }); });</pre>

购票	<pre> app.post('/film', function(req, res) { console.log(req.query.id); film_name = req.query.id; //console.log(req.body.name); //console.log(req.body.tel); res.sendFile(__dirname + "/" + "cinema.html" }); app.post('/cinema', function(req, res) { console.log(req.query.id); cinema_name = req.query.id; //console.log(req.body.name); //console.log(req.body.tel); res.sendFile(__dirname + "/" + "film.html"); }); </pre>
更新数据库	<pre> app.post('/insert_data', function(req, res) { connection.connect(); var addSql = 'INSERT INTO reservation(id,name,cinema,film) VALUES(?,?,?,?)' var addSqlParams = [myid,myname, cinema_name,film_name]; //增 connection.query(addSql,addSqlParams,function (err, result) { if(err){ console.log('[INSERT ERROR] - ',err.message); return; } // 生成订单信息 console.log('-----INSERT-----'); }); } </pre>
<p>模板文件 .ejs :</p> <p>这是前端网页的内容，因为与服务器关系密切，暂且放在这里介绍。</p> <p>位于 app.js 和 continue.ejs 文件中</p>	<pre> res.render('reservation.ejs', { user_name: myname, c_name: cinema_name, f_name: film_name }); </pre> <pre> <div class="signin-head"></div> <h3>Hi <%= user_name %>,You login successfully</h3> <h4 > Click here to Continue</h4> </pre>