Slice of Pie

Dunno

November 27, 2012

1 Preface

Contents

2 Abstract

3 Scrum

3.1 Definition of Done

3.1.1 General

- Must be reviewed and accepted by another team member
- Story must be completed from beginning to end

3.1.2 Documentation

- Must be in digital form

3.1.3 Program

- Proper documentation has been written. This includes full documentation for public methods
- Tests have been written and they they do not fail
- New code must not break previously written tests

4 Software Analysis

4.1 System Requirements

4.2 Use Cases

4.3 Domain Model

4.4 System Sequence Diagrams

4.5 FURPS+

5 Software Design

5.1 Class Diagram

5.2 Interaction Diagram

6 Database Design

We use a database on our server, to keep track of who owns which files, who has access and who made what changes to them.
Our relational database contains eight tables:
User describes a user. Email is used as primary key
File describes a file on the server. We use serverpath to describe the path to the folder where the file is located, and name as file name. We don't keep track of folders. By specifying the path to the folder of the file, describing folders become unnecessary. Only downside is that we are unable to handle empty folders as an empty folder is not described by any files.
FileMetaData holds meta date for a file such as resolution for pictures.
FileInstance is a relation between User and File. This describes the local path for a specific user, which allows different users to store their copy of a file, in different locations.
Change is used to keep track of who changed what in which file at what time.
Project holds the title of the project.
ProjectHasFile keeps track of which files a project references.
UserHasProject keeps track of which projects a user has.

6.1 Merge Policy

6.2 Client GUI

6.3 Web GUI

7 Software Architecture

7.1 Architecture Analysis

7.2 Scenarios

7.3 Factor Tables

7.4 Logical Views

7.5 Deployment Views

8 Testing

8.1 Test strategy and test results

8.1.1 Definition of thorough testing

- All public methods must be tested
- All fail scenarios must be tested
- Boundary testing

9 Conclusion

9.1 Reflection Upon Result

10 Appendix

10.1 MySQL Schema

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

CREATE SCHEMA IF NOT EXISTS `SliceOfLife` DEFAULT CHARACTER SET
latin1 COLLATE latin1_swedish_ci ;
USE `SliceOfLife` ;

-- -----------------------------------------------------
-- Table `SliceOfLife`.`Project`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `SliceOfLife`.`Project` (
  `id` VARCHAR(45) NOT NULL ,
  `title` VARCHAR(400) NOT NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `SliceOfLife`.`FileMetaData`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `SliceOfLife`.`FileMetaData` (
  `id` INT NOT NULL ,
  `type` VARCHAR(400) NULL ,
  `value` VARCHAR(400) NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `SliceOfLife`.`File`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `SliceOfLife`.`File` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(400) NOT NULL ,
  `serverpath` VARCHAR(400) NOT NULL ,
  `deleted` TINYINT UNSIGNED NULL DEFAULT 0 ,
  `Project_id` VARCHAR(45) NOT NULL ,
  `FileMetaData_id` INT NOT NULL ,
  PRIMARY KEY (`id`) ,
  INDEX ‘fk_File_Project1_idx‘ (‘Project_id‘ ASC) ,
  INDEX ‘fk_File_FileMetaData1_idx‘ (‘FileMetaData_id‘ ASC) ,
  CONSTRAINT ‘fk_File_Project1‘
    FOREIGN KEY (‘Project_id‘ )
    REFERENCES ‘SliceOfLife‘.‘Project‘ (‘id‘ )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT ‘fk_File_FileMetaData1‘
    FOREIGN KEY (‘FileMetaData_id‘ )
    REFERENCES ‘SliceOfLife‘.‘FileMetaData‘ (‘id‘ )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table ‘SliceOfLife‘.‘User‘
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS ‘SliceOfLife‘.‘User‘ (
  ‘email‘ VARCHAR(45) NOT NULL ,
  PRIMARY KEY (‘email‘) ,
  UNIQUE INDEX ‘email_UNIQUE‘ (‘email‘ ASC) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table ‘SliceOfLife‘.‘FileInstance‘
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS ‘SliceOfLife‘.‘FileInstance‘ (
  ‘id‘ INT UNSIGNED NOT NULL ,
  ‘File_id‘ INT UNSIGNED NOT NULL ,
  ‘User_email‘ VARCHAR(45) NOT NULL ,
  ‘path‘ VARCHAR(400) NOT NULL ,
  ‘deleted‘ TINYINT UNSIGNED NULL DEFAULT 0 ,
  PRIMARY KEY (‘id‘, ‘File_id‘, ‘User_email‘) ,
  INDEX ‘fk_FileInstance_File1_idx‘ (‘File_id‘ ASC) ,
  INDEX ‘fk_FileInstance_User1_idx‘ (‘User_email‘ ASC) ,
  CONSTRAINT ‘fk_FileInstance_File1‘
    FOREIGN KEY (‘File_id‘ )
    REFERENCES ‘SliceOfLife‘.‘File‘ (‘id‘ )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT ‘fk_FileInstance_User1‘
    FOREIGN KEY (‘User_email‘ )
    REFERENCES ‘SliceOfLife‘.‘User‘ (‘email‘ )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table ‘SliceOfLife‘.‘Change‘
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS ‘SliceOfLife‘.‘Change‘ (
  ‘id‘ INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  ‘User_Project_Title‘ VARCHAR(400) NOT NULL ,
  ‘File_id‘ INT UNSIGNED NOT NULL ,
  ‘User_email‘ VARCHAR(45) NOT NULL ,
  ‘timestamp‘ BIGINT NULL ,
  ‘change‘ TEXT NULL ,
  PRIMARY KEY (‘id‘, ‘User_Project_Title‘, ‘File_id‘, ‘User_email‘)
,
  INDEX ‘fk_Change_File1_idx‘ (‘File_id‘ ASC) ,
  INDEX ‘fk_Change_User1_idx‘ (‘User_email‘ ASC) ,
  CONSTRAINT ‘fk_Change_File1‘
    FOREIGN KEY (‘File_id‘ )
    REFERENCES ‘SliceOfLife‘.‘File‘ (‘id‘ )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT ‘fk_Change_User1‘
    FOREIGN KEY (‘User_email‘ )
    REFERENCES ‘SliceOfLife‘.‘User‘ (‘email‘ )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table ‘SliceOfLife‘.‘ProjectHasUser‘
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS ‘SliceOfLife‘.‘ProjectHasUser‘ (
  ‘Project_id‘ VARCHAR(45) NOT NULL ,
  ‘User_email‘ VARCHAR(45) NOT NULL ,
  PRIMARY KEY (‘Project_id‘, ‘User_email‘) ,
  INDEX ‘fk_ProjectHasUser_Project1_idx‘ (‘Project_id‘ ASC) ,
  INDEX ‘fk_ProjectHasUser_User1_idx‘ (‘User_email‘ ASC) ,
  CONSTRAINT ‘fk_ProjectHasUser_Project1‘
    FOREIGN KEY (‘Project_id‘ )
    REFERENCES ‘SliceOfLife‘.‘Project‘ (‘id‘ )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT ‘fk_ProjectHasUser_User1‘
    FOREIGN KEY (‘User_email‘ )
    REFERENCES ‘SliceOfLife‘.‘User‘ (‘email‘ )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;



SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```