KSU Student Portal

MSE Project Final Presentation

Javier Ramos Rodríguez

OUTLINE

- Introduction
- Implementation Phase
  - Phase 2 Design
  - Process
  - Implementation Overview
  - Phase 3 Implementation Design
  - Issues
  - Features
  - Demo
- Project Plan
- Test Plan
- Conclusions
- References

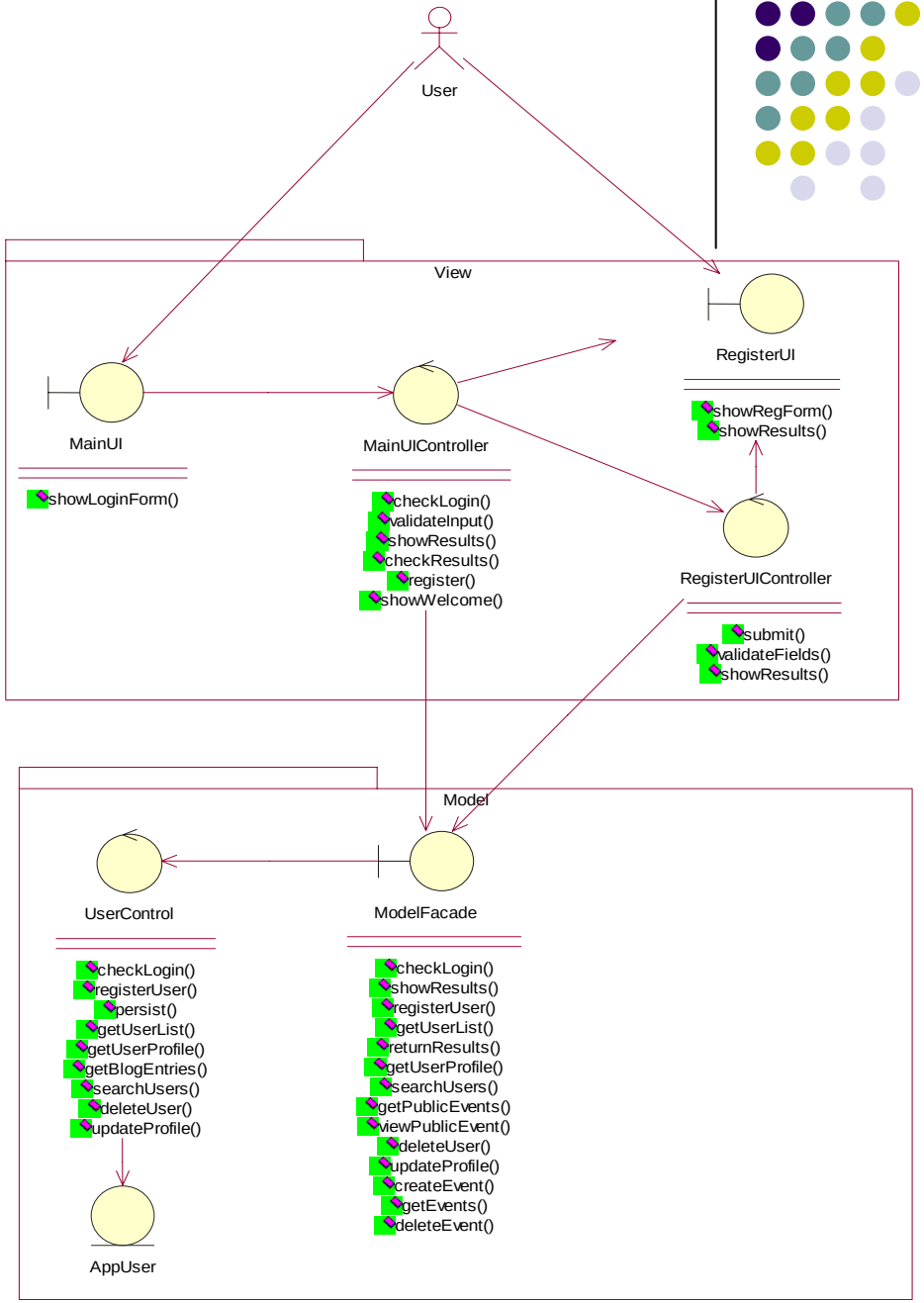
Introduction

- KSU Student Portal
  - Web Application
  - Data Bound
  - Heavyweight Approach
- Phase 3: Project Implementation
- Bottom-Up Approach
  - Data Entities
- Top-Down Approach
  - Facade Methods

Phase 2 Design

Use the Information provided by the Design and proceed with the system implementation.

More Details will be added to the application.



Process

- Phase 1 :

1. Requirements Analysis
2. Use Case Diagram
3. Use Case Specification
4. E-R Diagram

Process

- Phase 2:

1. Use Case Diagram Revision
2. Sequence & Collaboration Diagrams
3. Class Diagram
4. E-R Revision
5. UML Data Model
6. Relational Schema
7. UML Design Revision
8. Formal Specification

Process

- Phase 3:

1. Implement Relational Schema in Database
2. Create Entity Beans From Tables
3. Implement Relations Between the Entity Beans
4. Write initial EJB-QL Queries
5. Create Session Beans
6. Create Model Facade
7. Publish Methods
8. Test Methods

Implementation Overview

- There are more Session Beans than Control Classes from Design
- Load Balance
- Loosy Coupled
- Session Beans perform the business logic accessing to the Entity Beans
- Model Facade is the entry point to the system.
  - Model Controller
  - No Business logic, it relies on the EJBs
  - Session Facade, Business Delegate

Implementation Overview

- Entity Beans
  - AppUser
  - Blog, Blog Entry
  - Message
  - Article, Event, Link
  - Profile, Filter
  - Course, Language, Country, State

Implementation Overview

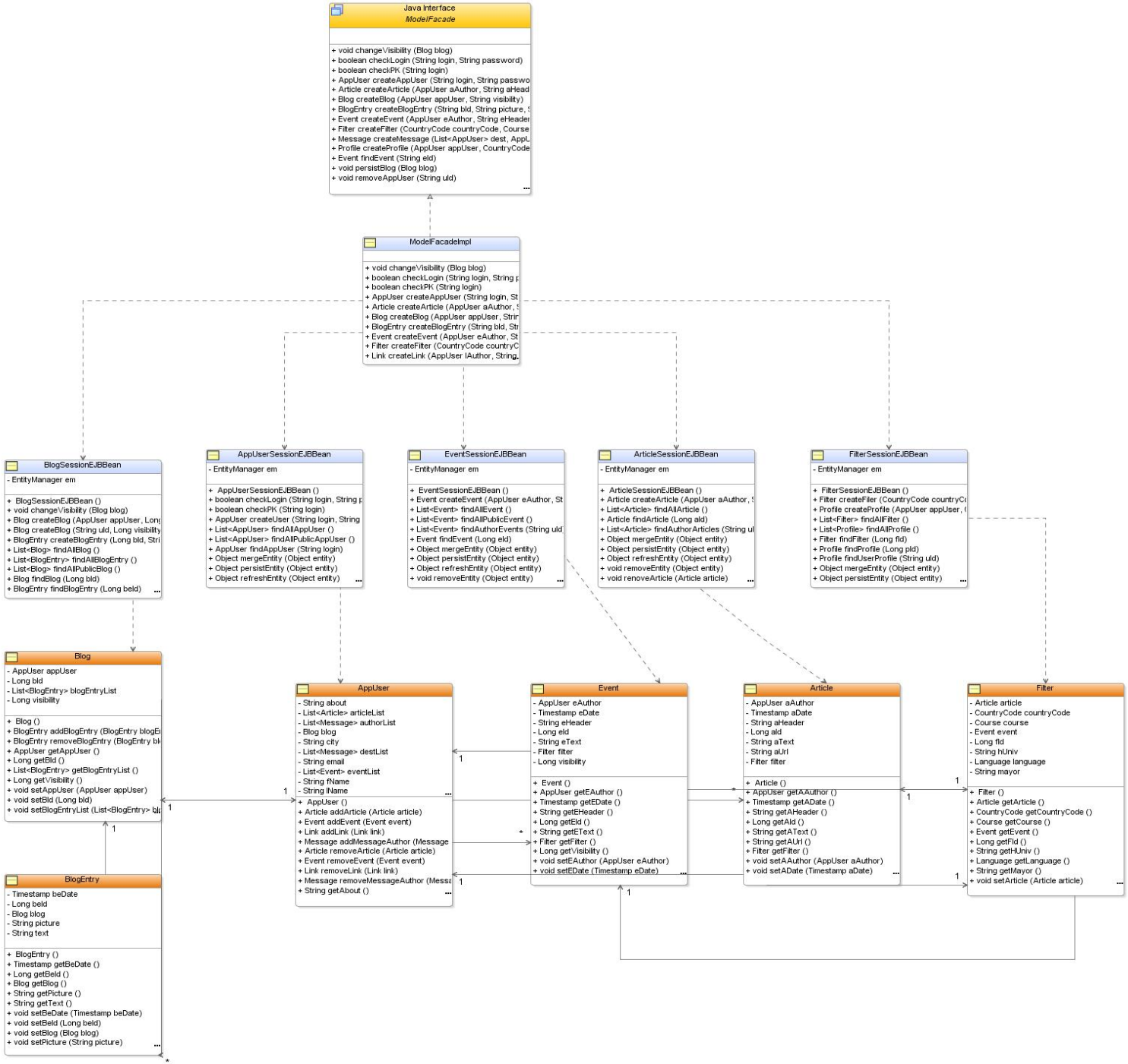
- Session Beans:
  - AppUserSessionBean
  - BlogSessionBean
  - MessageSessionBean
  - ArticleSessionBean, EventSessionBean, LinkSessionBean
  - FilterSessionBean
  - UtilSessionBean
- Model Facade

Implementation Design

Java Interface

ModelFacade

```
+ void changeVisibility (Blog blog)
+ boolean checkLogin (String login, String password)
+ boolean checkPK (String login)
+ AppUser createAppUser (String login, String password, String fName, String lName, String about, String city, String email, String picture, String quotes, String state, String street, String telf, String visibility)
+ Article createArticle (AppUser aAuthor, String aHeader, String aText, String aUrl, Filter filter)
+ Blog createBlog (AppUser appUser, String visibility)
+ BlogEntry createBlogEntry (String bId, String picture, String text)
+ Event createEvent (AppUser eAuthor, String eHeader, String eText, String visibility, Filter filter)
+ Filter createFilter (CountryCode countryCode, Course course, String hUniv, Language language, String mayor)
+ Message createMessage (List<AppUser> dest, AppUser mAuthor, String mText, String subject)
+ Profile createProfile (AppUser appUser, CountryCode countryCode, List<Course> courseList, String hUniv, List<Language> languageList, String mayor)
+ Event findEvent (String eId)
+ void persistBlog (Blog blog)
+ void removeAppUser (String uId)
+ void removeArticle (Article article)
+ void removeBlog (String bId)
+ void removeBlogEntry (String beId)
+ void removeEvent (Event event)
+ void removeMessage (Message message)
+ void updateAppUser (AppUser appUser)
+ void updateArticle (Article article)
+ void updateBlogEntry (BlogEntry blogEntry)
+ void updateEvent (Event event)
+ Filter updateFilter (Filter filter)
+ Profile updateProfile (Profile profile)
+ AppUser getAppUser (String uId)
+ List<AppUser> getAppUsers ()
+ Article getArticle (String aId)
+ List<Article> getArticles ()
+ List<Article> getAuthorArticles (String uId)
+ List<Event> getAuthorEvents (String uId)
+ List<Message> getAuthorMessages (String uId)
+ Blog getBlog (String bId)
+ List<Blog> getBlogs ()
+ List<CountryCode> getCountries ()
+ List<Course> getCourses ()
+ List<Message> getDestMessages (String uId)
+ List<Event> getEvents ()
+ Filter getFilter (String fId)
+ List<Filter> getFilters ()
+ List<Language> getLanguages ()
+ Message getMessage (String mId)
+ List<AppUser> getMessageDest (String mId)
+ List<Message> getMessages ()
+ Profile getProfile (String pId)
+ List<Profile> getProfiles ()
+ List<AppUser> getPublicAppUsers ()
+ List<Blog> getPublicBlogs ()
+ List<Event> getPublicEvents ()
+ List<State> getStates ()
+ Blog getUserBlog (String uId)
+ Profile getUserProfile (String uId)
```


Java Interface
ModelFacade
+ void changeVisibility (Blog blog)
+ boolean checkLogin (String login, String password)
+ boolean checkPK (String login)
+ AppUser createAppUser (String login, String passwo
+ Article createArticle (AppUser aAuthor, String aHead
+ Blog createBlog (AppUser appUser, String visibility)
+ BlogEntry createBlogEntry (String bId, String picture,
+ Event createEvent (AppUser eAuthor, String eHeader
+ Filter createFilter (CountryCode countryCode, Course
+ Message createMessage (List<AppUser> dest, AppU
+ Profile createProfile (AppUser appUser, CountryCode
+ Event findEvent (String eId)
+ void persistBlog (Blog blog)
+ void removeAppUser (String uId)
ModelFacadeImpl
+ void changeVisibility (Blog blog)
+ boolean checkLogin (String login, String p
+ boolean checkPK (String login)
+ AppUser createAppUser (String login, St
+ Article createArticle (AppUser aAuthor,
+ Blog createBlog (AppUser appUser, Strin
+ BlogEntry createBlogEntry (String bId, Str
+ Event createEvent (AppUser eAuthor, St
+ Filter createFilter (CountryCode countryC
+ Link createLink (AppUser lAuthor, String
BlogSessionEJBBean
- EntityManager em
+ BlogSessionEJBBean ()
+ void changeVisibility (Blog blog)
+ Blog createBlog (AppUser appUser, Long
+ Blog createBlog (String uId, Long visibility
+ BlogEntry createBlogEntry (Long bId, Stri
+ List<Blog> findAllBlog ()
+ List<BlogEntry> findAllBlogEntry ()
+ List<Blog> findAllPublicBlog ()
+ Blog findBlog (Long bId)
+ BlogEntry findBlogEntry (Long beId)
AppUserSessionEJBBean
- EntityManager em
+ AppUserSessionEJBBean ()
+ boolean checkLogin (String login, String p
+ boolean checkPK (String login)
+ AppUser createUser (String login, String
+ List<AppUser> findAllAppUser ()
+ List<AppUser> findAllPublicAppUser ()
+ AppUser findAppUser (String login)
+ Object mergeEntity (Object entity)
+ Object persistEntity (Object entity)
+ Object refreshEntity (Object entity)
EventSessionEJBBean
- EntityManager em
+ EventSessionEJBBean ()
+ Event createEvent (AppUser eAuthor, St
+ List<Event> findAllEvent ()
+ List<Event> findAllPublicEvent ()
+ List<Event> findAuthorEvents (String uId
+ Event findEvent (Long eId)
+ Object mergeEntity (Object entity)
+ Object persistEntity (Object entity)
+ Object refreshEntity (Object entity)
+ void removeEntity (Object entity)
ArticleSessionEJBBean
- EntityManager em
+ ArticleSessionEJBBean ()
+ Article createArticle (AppUser aAuthor,
+ List<Article> findAllArticle ()
+ Article findArticle (Long aId)
+ List<Article> findAuthorArticles (String uI
+ Object mergeEntity (Object entity)
+ Object persistEntity (Object entity)
+ Object refreshEntity (Object entity)
+ void removeEntity (Object entity)
+ void renoveArticle (Article article)
FilterSessionEJBBean
- EntityManager em
+ FilterSessionEJBBean ()
+ Filter createFiler (CountryCode countryC
+ Profile createProfile (AppUser appUser,
+ List<Filter> findAllFilter ()
+ List<Profile> findAllProfile ()
+ Filter findFilter (Long fId)
+ Profile findProfile (Long pId)
+ Profile findUserProfile (String uId)
+ Object mergeEntity (Object entity)
+ Object persistEntity (Object entity)
Blog
- AppUser appUser
- Long bId
- List<BlogEntry> blogEntryList
- Long visibility
+ Blog ()
+ BlogEntry addBlogEntry (BlogEntry blogE
+ BlogEntry removeBlogEntry (BlogEntry bl
+ AppUser getAppUser ()
+ Long getBId ()
+ List<BlogEntry> getBlogEntryList ()
+ Long getVisibility ()
+ void setAppUser (AppUser appUser)
+ void setBId (Long bId)
+ void setBlogEntryList (List<BlogEntry> bl
AppUser
- String about
- List<Article> articleList
- List<Message> authorList
- Blog blog
- String city
- List<Message> destList
- String email
- List<Event> eventList
- String fName
- String lName
+ AppUser ()
+ Article addArticle (Article article)
+ Event addEvent (Event event)
+ Link addLink (Link link)
+ Message addMessageAuthor (Message
+ Article removeArticle (Article article)
+ Event removeEvent (Event event)
+ Link removeLink (Link link)
+ Message removeMessageAuthor (Messa
+ String getAbout ()
Event
- AppUser eAuthor
- Timestamp eDate
- String eHeader
- Long eId
- String eText
- Filter filter
- Long visibility
+ Event ()
+ AppUser getEAuthor ()
+ Timestamp getEDate ()
+ String getEHeader ()
+ Long getEId ()
+ String getEText ()
+ Filter getFilter ()
+ Long getVisibility ()
+ void setEAuthor (AppUser eAuthor)
+ void setEDate (Timestamp eDate)
Article
- AppUser aAuthor
- Timestamp aDate
- String aHeader
- Long aId
- String aText
- String aUrl
- Filter filter
+ Article ()
+ AppUser getAAuthor ()
+ Timestamp getADate ()
+ String getAHeader ()
+ Long getAId ()
+ String getAText ()
+ String getAUrl ()
+ Filter getFilter ()
+ void setAAuthor (AppUser aAuthor)
+ void setADate (Timestamp aDate)
Filter
- Article article
- CountryCode countryCode
- Course course
- Event event
- Long fId
- String hUniv
- Language language
- String mayor
+ Filter ()
+ Article getArticle ()
+ CountryCode getCountryCode ()
+ Course getCourse ()
+ Event getEvent ()
+ Long getFId ()
+ String getHUniv ()
+ Language getLanguage ()
+ String getMayor ()
+ void setArticle (Article article)
BlogEntry
- Timestamp beDate
- Long beId
- Blog blog
- String picture
- String text
+ BlogEntry ()
+ Timestamp getBeDate ()
+ Long getBeId ()
+ Blog getBlog ()
+ String getPicture ()
+ String getText ()
+ void setBeDate (Timestamp beDate)
+ void setBeId (Long beId)
+ void setBlog (Blog blog)
+ void setPicture (String picture)

Issues

- Object Relationships
  - OO Schema = Relational Schema
- Lazy vs. Eager Fetching
- Handling Null Values (Filter)
- EJB-QL (Avoid Using SQL => Portability)

Features

- Distribute System
- Layers Pattern
- Database Independent: ORM + EJB-QL
- Safe (No Arrays) and clean code.
- New Technologies: EJB 3.0, Java 5
- Portable
- Pure OO
- Security: Encrypted passwords, code analyzers
- View: MVC, JSF + ADF Components, JavaScript, CSS, Multiple Languages…

Demo

Project Plan

- **What has been done**
  - Requirements Analysis: 100%
  - System Design: 100%
  - Implementation Design: 70%
  - Coding:
    - Database Layer: 100%
    - Model Layer: 85%
    - View Layer: 2%
    - Client layer: 0%

Project Plan

- What remains to be done
  - May: Finish Model, Start View
  - June: View (JSP pages)
  - July: Testing, Documentation, Enhance View

Test Plan

- System was build with testing in mind.
- Safe environment.
- Test Plan:
  - Unit Testing: *Junit*
  - Integration Tests
  - Other Tools (*FindBugs*)

Conclusions

- **Goals**
  - Mimic a real world enterprise application.
  - Full software Cycle.
  - Test New Technologies.
  - Learn J2EE Platform.
  - Solve problems using Javadoc and specifications.

Conclusions

- **EJB 3.0**
  - Easier to use than EJB 2.0.
  - Perform the most tedious work for you (network, security issues).
  - Standard.
  - Portable.
  - Immature, lack of documentation
  - Different interpretations of the specification.
  - Still difficult to use.

Conclusions

- **JSF**
  - Drag & Drop UI Design for Web Applications.
  - Easy to use.
  - MVC pattern.
  - Validation, Internationalization
  - Immature
  - MVC not as pure as with Struts

References

- www.uml.org
- www.rational.com
- Applying UML and Patterns - An Introduction to Object-Oriented Analysis and Design and the Unified Process, *Craig Larman*
- The Rational Unified Process: An Introduction (2nd Edition), *Philippe Kruchten*
- Java Server Faces (2004), *Hans Bergsten*