Team Logistics

Team Name

The X Factor

Changes in Logistics from Team Deliverable 2

There are no changes in logistics from Team Deliverable 2.

Success Criterion Most Difficult To Achieve

The most difficult success criterion for us is our original criteria 2: "Have what you promised to do done by the date you were assigned." — it's not always possible to complete your part of the work on time because of other classes possibly getting in the way.

Project Status Reports

Current State: On Schedule. We have not missed any deadlines, are currently complete with all assigned tasks, and are making sure we know where to go with our project.

Issues Encountered:

1. We didn't list three issues encountered in team deliverable 2.

There's no major way to fix this for team deliverable 2, but we can make sure to have three issues encountered listed in team deliverable 3. This was resolved by Brooke, as she made sure to verify that we had three issues listed in this team deliverable.

2. The project schedule from team deliverable 2 didn't show who is doing what.

In order to fix this issue, we have included a revised project schedule, which has been created by Ryan, as he identified which members are working on what tasks. The project schedule follows,

Project Schedule and Work Breakdown Structure:

- Complete Team Deliverable 2 10/19
- Begin work on design deliverable 11/5:
  - Brooke: Completed high level description of design, use cases, and analysis class change summary.
  - Hao, Leke, Ryan, Brass: Created UML diagrams
- Complete design deliverable 11/9:
  - Brooke: Rendered deliverable
- Begin work on implementation 11/9:
  - Hao: Research on software tools
  - Leke: Implement in tools based off Hao's suggestions
  - Brass: Ensure that software is designed to specs
- Continue working on implementation 11/16
  - Hao: Continue research
  - Leke: Continue Implementation
  - Brass: Check that current software is up to specs
  - Brooke, Ryan: Provide support as needed
- Start working on Team Presentation 11/16:
  - Hao, Leke, Brass: Give technical information
  - Ryan, Brooke: Clean up, check presentation quality
- Continue working on implementation & Team Presentation 11/23:
  - Hao, Leke, Brass: Work on software, provide support on Presentation as needed
  - Brooke, Ryan: Work on presentation, provide support as needed
- Complete implementation 11/30:
  - Brass: Quality check on software
  - Hao, Leke, Brooke, Ryan: Test software
- Continue working on Team Presentation 11/30:
  - Hao, Leke, Brass: Finalize software details on presentation
  - Hao, Leke, Brass, Brooke, Ryan: Practice WIP team presentation
- Complete Team Presentation 12/3:
  - Hao, Leke, Brass, Brooke, Ryan: Practice presentation, quality check presentation

- Begin work on Team Deliverable 5 11/30:
  - Hao, Leke, Brass, Brooke, Ryan: Work on deliverable
- Continue working on Team Deliverable 5 12/7:
  - Hao, Leke, Brass, Brooke, Ryan: Work on deliverable
- Complete work on Team Deliverable 5 12/14:
  - Hao, Leke, Brass, Brooke, Ryan: Quality check on deliverable

3. We started team deliverable 3 a bit late.

In order to fix this, we made sure to work overtime in order to complete the deliverable on time and have it prepared for mini assignment 4. The entire team chipped in to resolve this issue.

We have described how team members have contributed to the deliverables in the new project schedule.

Project Deliverables

High Level Description of System Design

Unified Name Changer aims to unify the separate systems of RPI's preferred and legal name change systems, as well as provide a simple way to access the form to get your diploma changed after a legal name change. In order to do this, we will have a "starter" function that will call one of three possible options. Two options, legal and preferred name change, have a common interface. The third option, diploma reissue, will be unique, as it can only lead you to the form, not fill it out for you (as it is a physical form which must be mailed in.) The two name change options will access all the disparate RPI name change methods, and complete them all. This will result in the preferred name change system accessing all different interfaces (such as LMS, Roundcube, Submitty, and Webex) and changing the preferred name in all of them, while the legal name change system will craft all necessary emails given the name change documentation, and send notices of name change to all different departments at RPI. After the legal name change system crafts all the emails, the user will have to do follow-ups in their email inbox, but it kickstarts the process, saving the user time.

Use Case 1: Update Preferred Name in LMS

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update preferred name in LMS.

Other Students: Need to know how to refer to their classmates.

Professors: Need to know students in classes

Success Scenario

1. Student selects preferred name change in system.
2. Student logs into application with RCSID and password.
3. Student inputs preferred name.
4. System changes preferred name in LMS.

Extensions

4a: Student enters incorrect RCSID and/or password, student returns to step 2.

4b: Student misspells preferred name, return to step 3.

4c: LMS is down, use case fails.

Preconditions

Student is a student at RPI.

Student has an LMS account.

Minimal Guarantees

Student is still a student.

Student still has an LMS account.

Success Guarantees

Student's name is updated in LMS.

Use Case 2: Update Preferred Name in RPI's Directory

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update preferred name in RPI's directory.

Professors: Need to know what to call students in class.

Administration: Need to know student preferred name in order to properly address student.

Success Scenario

1. Student selects preferred name change in system.
2. Student logs into application with RCSID and password.
3. Student inputs preferred name.
4. System changes preferred name in RPI's directory.

Extensions

4a: Student enters incorrect RCSID and/or password, student returns to step 2.

4b: Student misspells preferred name, return to step 3.

4c: Directory name change page is down, use case fails.

Preconditions

Student is a student at RPI.

Student has a directory entry.

Minimal Guarantees

Student is still a student.

Student still has a directory entry.

Success Guarantees

Student's name is updated in RPI's directory.

Use Case 3: Update Preferred Name in RPI's Housing System

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update preferred name in RPI's housing system.

Administration: Need to know student preferred name in order to properly address student.

Success Scenario

1. Student selects preferred name change in system.
2. Student logs into application with RCSID and password.
3. Student inputs preferred name.
4. System changes preferred name in RPI's housing system

Extensions

4a: Student enters incorrect RCSID and/or password, student returns to step 2.

4b: Student misspells preferred name, return to step 3.

4c: Housing system name page is down, use case fails.

Preconditions

Student is a student at RPI.

Student is in RPI's housing system.

Minimal Guarantees

Student is still a student.

Student is still in RPI's housing system.

Success Guarantees

Student's name is updated in RPI's housing system.

Use Case 4: Update Preferred Name in Submitty

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update preferred name in Submitty.

Other Students: Need to know how to refer to their classmates.

Professors: Need to know what to call student.

Success Scenario

1. Student selects preferred name change in system.
2. Student logs into appliation with RCSID and password.
3. Student inputs preferred name into application.
4. Application accesses Submitty's name update function and updates name.

Extensions

4a: Student enters incorrect RCSID and/or password, student returns to step 2.

4b: Student enters incorrect preferred name, student returns to step 3.

4c: Submitty is down, use case fails.

Preconditions

Student is a student.

Stu-dent has a Submitty account.

Minimal Guarantees

Student is still a student.

Student still has a Submitty account.

Success Guarantees

Student is still a student.

Student's name is updated in Submitty.

Use Case 5: Update Preferred Name in WebEx

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update preferred name in WebEx.

Other Students: Need to know how to refer to their classmates.

Professors: Need to know what to call student.

Success Scenario

1. Student selects preferred name change in system.
2. Student logs into application with RCSID and password.
3. Student inputs preferred name into application.
4. System inputs preferred name into WebEx profile.

Extensions

4a: Student enters incorrect RCSID and/or password, student returns to step 2.

4b: Student enters incorrect preferred name, student returns to step 3.

4c: WebEx is down, use case fails.

Preconditions

Student is a student.

Student has a WebEx account.

Minimal Guarantees

Student is still a student.

Student still has a WebEx account.

Success Guarantees

Student is still a student.

Student's name is updated in WebEx.

Use Case 6: Update Preferred Name in Email

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update preferred name in email.

Other Students: Need to know how to refer to their classmates.

Professors: Need to know what to call student.

Clubs: Need to know best name to refer to student.

Fraternities/Sororities: Need to know best name to refer to student.

Success Scenario

1. Student selects preferred name change in system.
2. Student logs into application with RCSID and password.
3. Student inputs preferred name into application.
4. System inputs preferred name into Roundcube.

Extensions

4a: Student enters incorrect RCSID and/or password, student returns to step 2.

4b: Student enters incorrect preferred name, student returns to step 3.

4c: Roundcube is down, use case fails.

Preconditions

Student is a student.

Student has an email address.

Minimal Guarantees

Student is still a student.

Student still has an email address.

Success Guarantees

Student is still a student.

Student's name is updated in Roundcube.

Use Case 7: Update Legal Name with Registrar

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update legal name in administrative database.

Registrar: Updates legal name.

Other Administrators: Need to know legal name for financial aid, bursar, etc.

Professors: Need to know what to call student.

Success Scenario

1. Student selects legal name change in system.
2. Student logs into application with RCSID and password.
3. Student uploads legal documentation detailing name change into system.
4. Student inputs legal name into system.
5. System sends automated email to registrar.
6. Registrar verifies legal documentation.
7. Registrar updates name in administrative database.

Extensions

5a: Student enters incorrect RCSID and/or password, student returns to step 2.

6a: Student provides insufficient/incorrect documentation, registrar notifies student, student returns to step 3.

6b: Registrar doesn't recieve email, use case fails.

Preconditions

Student is a student.

Minimal Guarantees

Student is still a student.

Success Guarantees

Student is still a student.

Student's name is updated in updated in administrative database.

Use Case 8: Update Legal Name with Campus Card Office

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update legal name on ID card.

Campus Card Office: Updates name on ID card, issues new ID card.

Success Scenario

1. Student selects new ID card in system.
2. Student is linked to campus card name change form.
3. Student completes form, including legal documentation.
4. Campus card office recieves and processes name change request form.
5. Campus card office prints new ID card
6. Campus card office alerts Student via email.

Extensions

3a: Form is down, use case fails.

4a: Student provides incorrect information on form, campus card office does not approve form, return to step 3.

Preconditions

Student is a student.

Student has legally changed their name with the registrar.

Minimal Guarantees

Student is still a student.

Success Guarantees

Student is still a student.

Student receives a new ID card.

Use Case 9: Update Legal Name with RPI Health Insurance

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update legal name on CDPHP insurance card.

RPI Health Center: Updates name in RPI's health insurance database.

CDPHP: Updates name on CDPHP insurance card after RPI Health Center updates health insurance database.

Success Scenario

1. Student selects legal name change in system.
2. Student logs into application with RCSID and password.
3. Student uploads legal documentation detailing name change into system.
4. Student inputs legal name into system.
5. System sends automated email to RPI Health Center.
6. RPI Health Center verifies legal documentation.
7. RPI Health Center updates name in Health Center database.
8. CDPHP updates name in CDPHP's database.

Extensions

5a: Student inputs incorrect login information, return to step 2.

6a: Student provides insufficient/incorrect documentation, RPI Health Center notifies student, student returns to step 3.

6b: RPI Health Center doesn't recieve email, use case fails.

Preconditions

Student is a student.

Minimal Guarantees

Student is still a student.

Success Guarantees

Student is still a student.

Student's name is updated in updated in RPI Health Center Database.

Student's name is updated on their CDPHP insurance card.

Use Case 10: Update Legal Name in DotCIO Ticketing System

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update legal name in DotCIO ticketing system.

DotCIO Support Center: Updates name in DotCIO ticketing system.

Success Scenario

1. Student selects legal name change in system.
2. Student logs into application with RCSID and password.
3. Student uploads legal documentation detailing name change into system.
4. Student inputs legal name into system.
5. Student inputs phone number.
6. DotCIO Support Center receives ticket.
7. DotCIO Support Center delegates staff member to handle ticket.
8. DotCIO Support Center verifies documentation
9. DotCIO Support Center changes student's legal name in ticketing system.

Extensions

6a: Student inputs incorrect login information, return to step 2.

6b: Ticketing system is down, use case fails.

8a: Student provides insufficient/incorrect documentation, DotCIO Support Center notifies student, student returns to step 3.

Preconditions

Student is a student.

Minimal Guarantees

Student is still a student.

Success Guarantees

Student is still a student.

Student's name is updated in DotCIO ticketing system.

Use Case 11: Update Legal Name in WebEx

Primary Actor

Student.

Stakeholders and Interests

Student: Wants to update legal name in WebEx.

DotCIO Support Center: Updates name in WebEx.

Success Scenario

1. Student selects legal name change in system.
2. Student logs into application with RCSID and password.
3. Student uploads legal documentation detailing name change into system.
4. Student inputs legal name into system.
5. Student inputs phone number.
6. DotCIO Support Center receives ticket.
7. DotCIO Support Center delegates staff member to handle ticket.
8. DotCIO Support Center verifies documentation
9. DotCIO Support Center changes student's legal name in WebEx.

Extensions

6a: Student inputs incorrect login information, return to step 2.

6b: Ticketing system is down, use case fails.

8a: Student provides insufficient/incorrect documentation, DotCIO Support Center notifies student, student returns to step 3.

Preconditions

Student is a student.

Minimal Guarantees

Student is still a student.

Success Guarantees

Student is still a student.

Student's name is updated in WebEx.

Use Case 12: Reissue Diploma to Graduated Student

Primary Actor

Former student.

Stakeholders and Interests

Former student: Wants reissued diploma.

Registrar: Needs to order new diploma.

Diploma Printing Company: Need to know name to print on new diploma, ships diploma to student.

Success Scenario

1. Student selects diploma reissue in system.
2. Student is provided with diploma reorder form.
3. Student fills out diploma reorder form.
4. Student notarizes form.
5. Student returns form, old diploma, and payment to Registrar.
6. Registrar receives form, old diploma, and payment.
7. Registrar confirms form is filled out properly and notarized.
8. Registrar puts in order for new diploma.
9. Printing company prints diploma.
10. Printing company sends diploma to student.

Extensions

6a. Form is lost in mail, use case fails.

6b. If student doesn't return diploma, Registrar informs student. Either student returns to step 6, or if student still doesn't return diploma, Registrar continues and informs student that new diploma will be marked as duplicate.

7a: Student fills out form incorrectly, Registrar informs former student, return to step 3.

7b: Student doesn't properly notarize form, Registrar informs former student, return to step 4.

Preconditions

Student has graduated.

Student has received a diploma.

Minimal Guarantees

Student was a student.

Success Guarantees

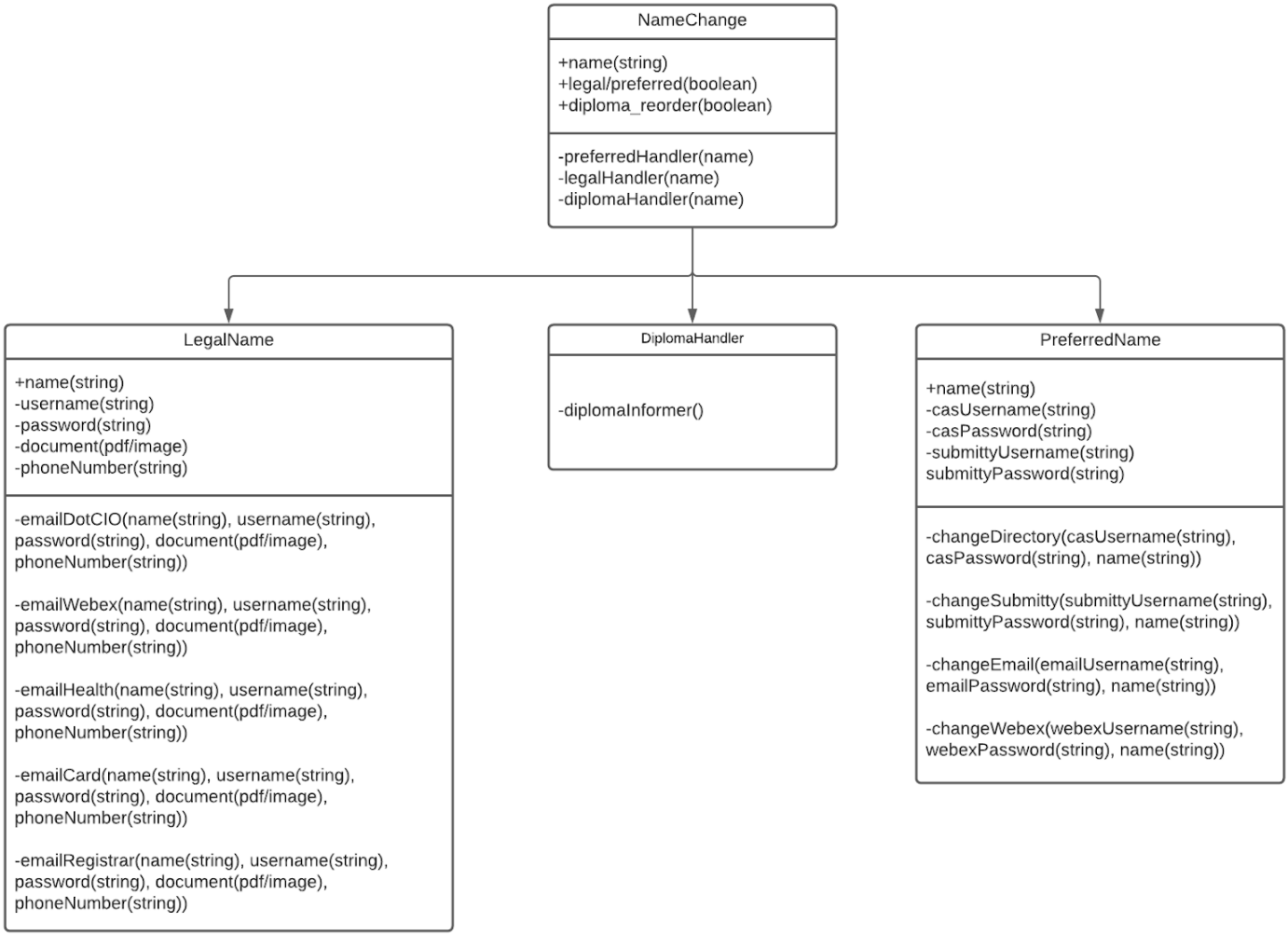
Student was a student.

Student receives updated diploma.

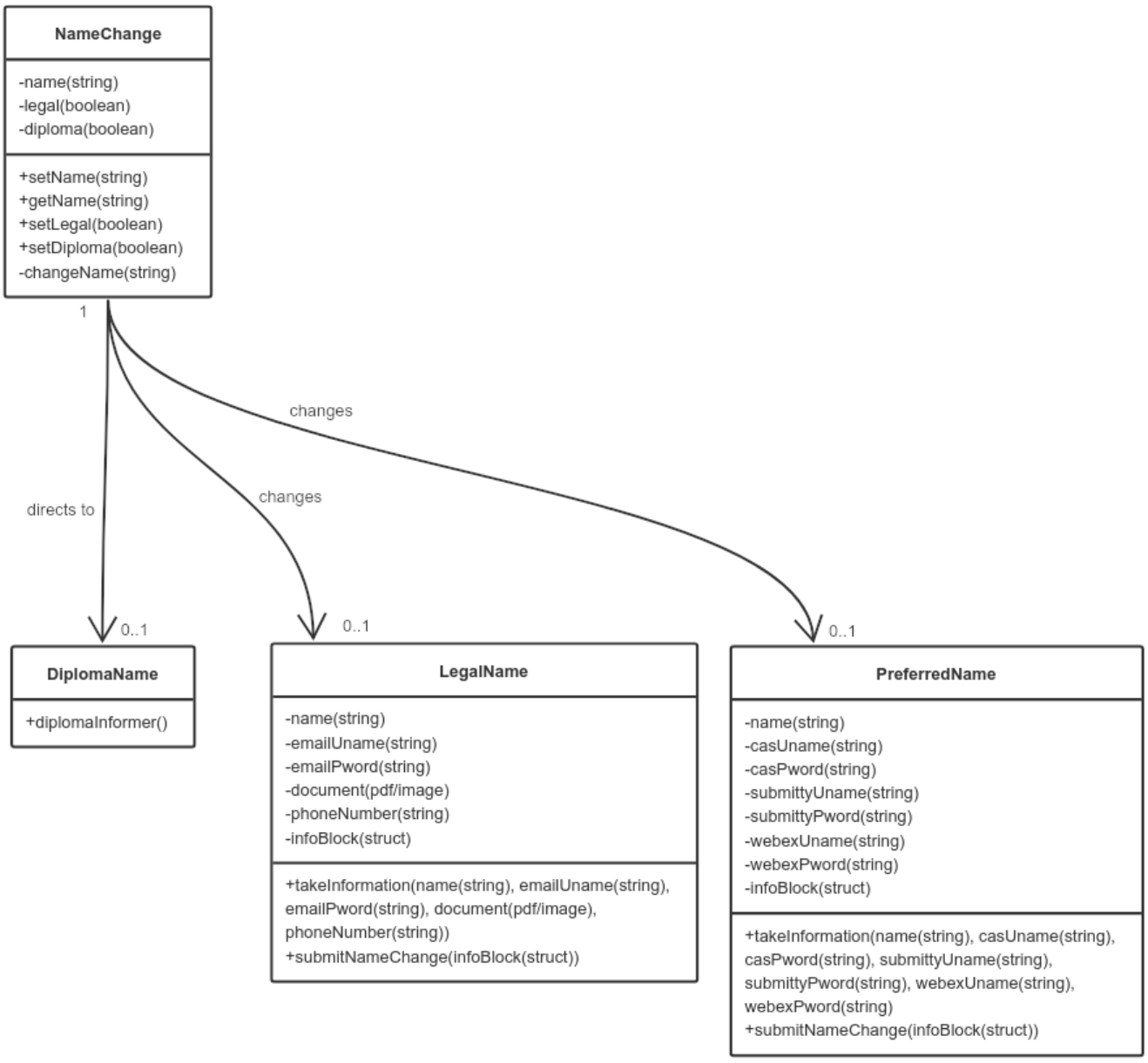
Summary of How Analysis Classes have Changed

We changed from having public variables to having private variables with public setters/getters. This is to better protect variables from being changed unintentionally where they shouldn't be. Additionally, we changed from having three different functions that would be called to one function that would use booleans to determine which class to call. This change was implemented due to the fact that we already had these booleans, so we were reducing complexity while maintaining feature parity. We also changed the Legal and Preferred Name Analysis Classes, adding a infoBlock structure variable to both Legal and Preferred Name, and adding webex username and password variables to preferred name. We changed how they act as well, with both having the same functions. This was informed by the design, but did change the analysis.

Original Analysis Class



New Analysis Class



Design Class, including UML Notes