**STUDENT EVENT WEBSHOP**

– Software Engineering Project –

**PROJECT PROGRESS REPORT**

**Team: Attila Gonczi (434519)**

**Jiening Wen (439947)**

**Dimitar Kolev (431646)**

**Arya Nawing (477490)**

**Lecturer: Eddy de Rooij**

**Version 1**

**Date 16/09/2009**

# Development Process Information

The Event Webshop project is currently in the last phase of its first development cycle (Cycle 1). According to the development state of this cycle, the deadline of finalizing Cycle 1 was modified in agreement with Mr. Eddy de Rooij, the director of Inholland Softwarehouse, and it is extended until 9 November 2009. By this date the Student Event Webshop team expects to complete all objectives set for Cycle 1.

|  |  |  |
| --- | --- | --- |
| Planned objectives | Competed (sub)tasks | State / Actions needed |
| Project Plan | Accepted by supervisor | Done |
| Version control | Project is maintained at Google code repository  All developers have TortoiseSVN installed for uploading/downloading revisions.  Project can be anonymously checked out at:  <http://webshopinholland.googlecode.com/svn/trunk> |  |
| All core features and functionalities of this application should be clearly defined. | Core features:  Front-end   1. Product display pages  * Top 10 * Search by topic * Search by institute * Search by location * Details view  1. Order handling  * Shopping cart * Checkout pages * Confirmation emails  1. Customer (buyer) account administration  * Signup * Data admin * Order admin * Purchase overview  1. Institute signup request 2. Payments using PayPal   Back-End   1. Application administration  * Adding default even properties * User administration (institutes and students)  1. Administration for Institutes  * Defining new events * Controlling event application states * Checking payments | Done |
| Define PayPal Accounts in Development Environment | Defined accounts:   * 5 business account are defined * 5 user accounts are defined   All developers have the necessary info for accessing the sandbox account  Virtual payments are functioning within sandbox  Different payment constructions are studied | To be done:  Check if provided API is suitable for the application, and if not, write custom classes for submitting and retrieving info to/from PayPal  Check how encryption/decryption of data communication between the system and PayPal is possible. |
| Define Database Structure | Database definition:   1. ER diagram is done 2. Event custom data tables in SQL Server 2005  * Tables are defined and normalized * Foreign key constraints are defined | To be done:   * Define default values * Define cascading and custom referential integrity logic |
| Populate Database with test data |  | To be done:  Write scripts that inserts test data into the database   1. Insert data from bottom-up in relation with foreign-key constraints |
| Test Database |  | To be done:   1. Test insertion of data  * Check correct default values * Check how null values are handled (required or not) * Test foreign-key integrity when insertion  1. Test updates of data  * Test tables that has no surrogate primary key(s)  1. Test foreign-key integrity when deletion  * Check if all allowed deletions are permitted by referential integrity constraints * Check if all disallowed deletions are prohibited |
| Design Basic front-end Interfaces | 1. Master page 2. Layout - done 3. Login - done 4. Search bar - done 5. Menu - done 6. Index page 7. Event summary template - done 8. Top 10 list - done 9. Event template - done 10. List style – done 11. Check out page (need to change layout later if time allows) 12. Personal info - done 13. Payment method – done 14. Confirmation - done   Note:  Even though event template is listed in different pages, it will be different for each page, because the way in which events will be presented on different pages will also be different. | To be done:   1. Event list page    1. Event template yet    2. List style 2. Event detail page    1. Event detail template   Adjust:  Check out page is partially done because the layout still has to be changed. |
| Create Data Access Layer |  | Still to be done |
| Implement design of front-end interfaces | Definition of interfaces in asp.net pages are done | To be done:  Create underlying programming |
| Assemble, debug and test the application |  | Postponed until week 10 |
| Presentation of results to Supervisor |  | Postponed until first week of Period 2 |
| Collect feedback from colleagues and supervisor |  | Postponed until first week of Period 2 |