# A stage – extending Wiki features (total 50 points)

Team Name: WikiWiki

Team Leader: Ryan Guard

Submission Date: 04/19/2018

# Integrity

(O) We (members of this team) understand that this project deliverable is for our understanding of SE and for our career development, so we did our best to make this assignment to our advantage and in high quality. We deducted points accordingly if we don’t have correct and high-quality results.

( ) We used the following teams’ results, or we shared our results with the following teams.

Team \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ because \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Delegation or self-grading

(O) We want the instructor use our grading results, and We certify that we graded as accurate as possible to the best of our knowledge.

(O) We graded ourselves.

( ) Team \_\_\_\_\_\_\_\_\_\_\_ graded this assignment.

( ) We don’t want to grade our project results, and we want the instructor to grade homework. So, we don’t check the rest of the grading rubrics.

* But we understand the instructor can deduct points depending on the quality of my submission.
* Even in this case, you should give points each of feature grading section.

**Total points earned: (50)/50**

**Plan and retrospect (15)/15**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Points earned | Proof | To the instructor |
| Schedule | (5) / 5 | * We planned a schedule ahead of time, and we followed the schedule. We also updated the schedule when we have a schedule change. * We included our (final) schedule in the deliverables. |  |
| Rules collection | (5) / 5 | * We discussed and collected SE rules that we could apply in this stage. * We included rules in the deliverables. |  |
| Retrospect | (5) / 5 | * We did a retrospect meeting to share what went wrong (and why) and what went well (and why). * We included retrospect meeting in the deliverables. |  |

**Extension of existing wiki system (35)/35 (a team leader aggregates from each implementer and grades)**

1. **Calculate N = 35 / x where x is the number of your team members.** 
   1. **N = 7, P = N / 3**
   2. **Each category is assigned to maximum P points.**
2. **Grade each feature to get maximum of N points.** 
   1. **For example, with 5 members in a team, N = 7. When a team member finished everything required, the member earns 100% to make 7 \* 100% = 7 points.**
   2. **When you don’t or can’t accomplish everything, you can give partial credit with your comment in the ‘To the instructor’ section explaining why. For example, if you finish about half of the implementation, you will grade 15% (30% are assigned for implementation but you implemented only 50%).**
   3. **Each feature is checked by other team member or a team leader.**
3. **The features implemented should not be trivial. The instructor expects each member use at least 15 hours for each feature. In the case that the implemented feature is too trivial, the instructor may give less points than you graded.**

**Feature 1: (7) / (7) points earned as (100)%/100% of the required work is done**

**Feature name: Document Archive system**

**The implementer: Ryan Guard**

**Checked by: Chris Groppe**

|  |  |  |  |
| --- | --- | --- | --- |
|  | % earned | Proof | To the instructor |
| Problem definition and requirements | (10)%/10% | * I defined the problem that I solved by clarifying the issue and your proposed solution. * I wrote a requirement in a form of user stories. |  |
| Design document | (20)%/20% | * I designed my feature that meets the requirement. |  |
| Implementation | (30)%/30% | * I implemented the feature according to the design. |  |
| Test | (20)%/20% | * I wrote effective doctests or unittests for verifying and validating my feature. |  |
| PIP library | (10)%/10% | * I created a PIP library for my feature and uploaded in the PIP respoitory. |  |
| Pydoc | (10)%/10% | * I used argparse to parse users input and display argument information. * I used pydoc for user documentation. |  |

**Feature 2: (7) / (7) points earned as (100)%/100% of the required work is done**

**Feature name: Amazon Webservices hosted Postgressql database**

**The implementer: Chris Groppe**

**Checked by: Ryan Guard**

|  |  |  |  |
| --- | --- | --- | --- |
|  | % earned | Proof | To the instructor |
| Problem definition and requirements | (10)%/10% | * I defined the problem that I solved by clarifying the issue and your proposed solution. * I wrote a requirement in a form of user stories. |  |
| Design document | (20)%/20% | * I designed my feature that meets the requirement. |  |
| Implementation | (30)%/30% | * I implemented the feature according to the design. |  |
| Test | (20)%/20% | * I wrote effective doctests or unittests for verifying and validating my feature. |  |
| PIP library | (10)%/10% | * I created a PIP library for my feature and uploaded in the PIP respoitory. | included documentation for implementing our database schema |
| Pydoc | (10)%/10% | * I used argparse to parse users input and display argument information. * I used pydoc for user documentation. |  |

**Feature 3: (7) / (7) points earned as (100)%/100% of the required work is done**

**Feature name: Document Recovery system**

**The implementer: Liam Tiemon**

**Checked by: Ryan Guard**

|  |  |  |  |
| --- | --- | --- | --- |
|  | % earned | Proof | To the instructor |
| Problem definition and requirements | (10)%/10% | * I defined the problem that I solved by clarifying the issue and your proposed solution. * I wrote a requirement in a form of user stories. |  |
| Design document | (20)%/20% | * I designed my feature that meets the requirement. |  |
| Implementation | (30)%/30% | * I implemented the feature according to the design. |  |
| Test | (20)%/20% | * I wrote effective doctests or unittests for verifying and validating my feature. |  |
| PIP library | (10)%/10% | * I created a PIP library for my feature and uploaded in the PIP respoitory. |  |
| Pydoc | (10)%/10% | * I used argparse to parse users input and display argument information. * I used pydoc for user documentation. |  |

**Feature 4: (7) / (7) points earned as (100)%/100% of the required work is done**

**Feature name: User interface to view and recover changes**

**The implementer: Elizabeth Gieske**

**Checked by: Ryan Guard**

|  |  |  |  |
| --- | --- | --- | --- |
|  | % earned | Proof | To the instructor |
| Problem definition and requirements | (10)%/10% | * I defined the problem that I solved by clarifying the issue and your proposed solution. * I wrote a requirement in a form of user stories. |  |
| Design document | (20)%/20% | * I designed my feature that meets the requirement. |  |
| Implementation | (30)%/30% | * I implemented the feature according to the design. |  |
| Test | (20)%/20% | * I wrote effective doctests or unittests for verifying and validating my feature. |  |
| PIP library | (10)%/10% | * I created a PIP library for my feature and uploaded in the PIP respoitory. | Comes in form of Display History.py |
| Pydoc | (10)%/10% | * I used argparse to parse users input and display argument information. * I used pydoc for user documentation. |  |

**Feature 5: (7) / (7) points earned as (100)%/100% of the required work is done**

**Feature name: Testing Suite for overall system**

**The implementer: Ronnie Hoover**

**Checked by: Ryan Guard**

|  |  |  |  |
| --- | --- | --- | --- |
|  | % earned | Proof | To the instructor |
| Problem definition and requirements | (10)%/10% | * I defined the problem that I solved by clarifying the issue and your proposed solution. * I wrote a requirement in a form of user stories. |  |
| Design document | (20)%/20% | * I designed my feature that meets the requirement. |  |
| Implementation | (30)%/30% | * I implemented the feature according to the design. |  |
| Test | (20)%/20% | * I wrote effective doctests or unittests for verifying and validating my feature. |  |
| PIP library | (10)%/10% | * I created a PIP library for my feature and uploaded in the PIP respoitory. |  |
| Pydoc | (10)%/10% | * I used argparse to parse users input and display argument information. * I used pydoc for user documentation. |  |