|  |  |  |
| --- | --- | --- |
| Use Case Name: | *View Assignment List* | |
| Scenario: | Melihat list assignment | |
| Trigger Event: | *Actor* ingin melihat assignment list | |
| Brief Description: | *Actor* dapat melihat list assignment sesuai dengan kelasnya | |
| Actor: | Siswa, Guru | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | Ada assignment yang ingin dilihat *actor* | |
| Postcondition: | Informasi *assignment* ditampilkan kepada *actor* | |
| Flow of Activity: | Actor | System |
| 1. Klik *class* yang di-inginkan  2. Klik menu *assignment* | 1.1 Redirect ke menu *class*  2.1 Menampilkan list *assignment* |
| Exception Condition: | - | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *View Assignment Question* | |
| Scenario: | Melihat *assignment* *question* | |
| Trigger Event: | *Actor* ingin melihat *assignment* *question* | |
| Brief Description: | *Actor* dapat melihat *assignment* *question* | |
| Actor: | Siswa, Guru | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | Ada *assignment* *question* yang ingin dilihat *actor* | |
| Postcondition: | Informasi *assignment* *question* ditampilkan kepada *actor* | |
| Flow of Activity: | Actor | System |
| 1. Klik *class* yang diinginkan  2. Klik menu *assignment*  3. Klik *download* *assignment*  4. Simpan *download* *assignment* | 1.1 Redirect ke menu *class*  2.1 Menampilkan list *assignment*  3.1 Sistem melakukan proses *download* *assignment* yang dipilih |
| Exception Condition: | - | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *Create Assignment Question* | |
| Scenario: | Membuat *assignment* baru | |
| Trigger Event: | *Actor* ingin membuat *assignment* | |
| Brief Description: | *Actor* dapat membuat *assignment* sesuai dengan kelasnya | |
| Actor: | Guru | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | Guru ingin membuat *assignment* baru | |
| Postcondition: | *Assignment* baru akan tersimpan dalam *database* *system* | |
| Flow of Activity: | Actor | System |
| 1. Klik *create new asignment*  2. Guru *input* *assignment* baru | 1.1 Menampilkan *form* untuk *add assignment*  2.1 *System* melakukan verifikasi seluruh parameter yang di-input *Actor*.  2.2 *System* menyimpan data yang di input ke *database*.  2.3 Redirect ke halaman *Dashboard* |
| Exception Condition: | 2.1. Notifikasi error karena karena data yang di input tidak sesuai. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *Update Assignment Question* | |
| Scenario: | *Update* *assignment* | |
| Trigger Event: | *Actor* ingin mengupdate *assignment* | |
| Brief Description: | *Actor* bisa mengupdate *assignment* | |
| Actor: | Guru | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | *Assignment* yang sudah dibuat ada perubahan | |
| Postcondition: | *Assignment* yang telah diperbaharui akan tersimpan pada database system | |
| Flow of Activity: | Actor | System |
| 1. Klik update pada *assignment* *list* yang ingin di update  2. Guru *input assignment* dengan data baru | 1.1 Menampilkan *form* untuk *edit assignment*  2.1 *System* melakukan verifikasi seluruh parameter yang di-input *Actor*.  2.2 *System* menyimpan data yang di input ke *database*.  2.3 Redirect ke halaman *Dashboard* |
| Exception Condition: | 2.1. Notifikasi error karena karena data yang di input tidak sesuai. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *Upload Assignment Answer* | |
| Scenario: | *Upload assignment* | |
| Trigger Event: | *Actor* ingin mengupload jawaban *assignment* | |
| Brief Description: | *Actor* bisa mengupload jawaban *assignment* | |
| Actor: | Siswa | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | Siswa ingin mengupload jawaban dari *assignment* yang ada | |
| Postcondition: | Jawaban a*ssignment* akan tersimpan pada *database* *system* | |
| Flow of Activity: | Actor | System |
| 1. Klik *add submission* pada salah satu *assignment* yang ada pada *assignment list*  2. Siswa *input assignment* jawaban pada *form* | 1.1 Menampilkan *form* untuk *add submission*  2.1 *System* melakukan verifikasi seluruh parameter yang di-input *Actor*.  2.2 *System* menyimpan data yang di input ke *database*.  2.3 Redirect ke halaman Dashboard |
| Exception Condition: | 2.1. Notifikasi error karena karena data yang di input tidak sesuai. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *View Assignment Submission* | |
| Scenario: | Melihat jawaban *assignment* siswa | |
| Trigger Event: | *Actor* ingin melihat jawaban *assignment* | |
| Brief Description: | *Actor* dapat melihat jawaban *assignment* yang di upload siswa sesuai dengan kelasnya | |
| Actor: | Guru | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | Guru ingin melihat jawaban *assignment* yang diupload oleh siswa | |
| Postcondition: | Informasi *assignment submission* ditampilkan kepada actor | |
| Flow of Activity: | Actor | System |
| 1. Klik *class* yang diinginkan  2. Klik menu *assignment*  3. Klik *show detail* | 1.1 Redirect ke menu *class*  2.1 Menampilkan *list assignment*  3.1 Sistem menampilkan *list* jawaban *assignment* seluruh siswa |
| Exception Condition: | - | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *View Assignment Submission History* | |
| Scenario: | Melihat jawaban *assignment history* siswa | |
| Trigger Event: | *Actor* ingin melihat *history* *submission* dari *assignment* | |
| Brief Description: | *Actor* dapat melihat *history* jawaban *assignment* yang di upload | |
| Actor: | Siswa | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | Siswa ingin melihat jawaban assignment yang pernah diupload oleh siswa | |
| Postcondition: | Informasi assignment submission history ditampilkan kepada actor | |
| Flow of Activity: | Actor | System |
| 1. Klik *class* yang diinginkan  2. Klik *menu assignment*  3. Klik *history submission* | 1.1 Redirect ke menu *class*  2.1 Menampilkan *list assignment*  3.1 Sistem menampilkan *list history* jawaban *assignment* siswa |
| Exception Condition: | - | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *View Score* | |
| Scenario: | Melihat *score* siswa | |
| Trigger Event: | *Actor* ingin melihat nilai siswa | |
| Brief Description: | *Actor* dapat melihat nilai siswa sesuai dengan *class*-nya | |
| Actor: | Siswa, Guru | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | Guru dan Siswa ingin melihat *score* siswa | |
| Postcondition: | Informasi *score* ditampilkan kepada *actor* | |
| Flow of Activity: | Actor | System |
| 1. Klik *class* yang diinginkan  2. Klik menu *score* | 1.1 Redirect ke menu *class*  2.1 Sistem menampilkan *score* siswa |
| Exception Condition: | - | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *Create Score* | |
| Scenario: | Menginput *score* siswa | |
| Trigger Event: | *Actor* ingin menginput *score* siswa | |
| Brief Description: | *Actor* dapat menginput *score* siswa sesuai dengan kelasnya | |
| Actor: | Guru | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | Guru ingin menginput *score* siswa | |
| Postcondition: | *Score* baru akan tersimpan dalam *database system* | |
| Flow of Activity: | Actor | System |
| 1. Klik *add score*  2. Guru *input score* siswa | 1.1 Menampilkan *form* untuk *add score*  2.1 *System* melakukan verifikasi seluruh parameter yang di-input Actor.  2.2 *System* menyimpan data yang di *input* ke *database*.  2.3 Redirect ke halaman *Dashboard* |
| Exception Condition: | 2.1. Notifikasi *error* karena karena data yang di input tidak sesuai. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *Update Score* | |
| Scenario: | Mengubah *score* siswa | |
| Trigger Event: | *Actor* ingin mengubah *score* siswa | |
| Brief Description: | *Actor* dapat mengubah *score* siswa | |
| Actor: | Guru | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | *Score* yang telah di *input* ada perubahan | |
| Postcondition: | *Score* baru akan tersimpan dalam *database system* | |
| Flow of Activity: | Actor | System |
| 1. Klik *update* pada *score* yang ingin di *update*  2. Guru update *score* siswa | 1.1 Menampilkan *form* untuk *edit score*  2.1 *System* melakukan verifikasi seluruh parameter yang di-*input* *Actor*.  2.2 *System* menyimpan data yang di input ke *database*.  2.3 Redirect ke halaman *Dashboard* |
| Exception Condition: | 2.1. Notifikasi *error* karena karena data yang di input tidak sesuai. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | *Delete Score* | |
| Scenario: | Menghapus *score* siswa | |
| Trigger Event: | *Actor* ingin menghapus *score* siswa | |
| Brief Description: | *Actor* dapat menghapus *score* siswa yang sudah di *input* | |
| Actor: | Guru | |
| Related Use Case: | - | |
| Stakeholders: | - | |
| Precondition: | *Score* yang telah di *input* sudah tidak relevan | |
| Postcondition: | *Score* akan dihapus dalam *database system* | |
| Flow of Activity: | Actor | System |
| 1. Klik *delete* pada *score* yang ingin di *delete*  2. Guru klik konfirmasi | 1.1 Menampilkan *form* konfirmasi  2.1 *System* menghapus data di *database*.  2.2 Redirect ke halaman *Dashboard* |
| Exception Condition: | - | |