SW Engineering CSC 648/848 - Milestone 4

SecondChance: The Eco-Friendly Rental Platform!

CSC 648 - Section 01 - Team 05
11/28/2024

Team Members & Roles

Student Name	Student's Role
Parth Desai	Team Lead / PM
Pedro Grande	Front-end Developer
Charvi Sharma	Scrum Master
Josue Hernandez	Git Master
Andre Velarde	Back-end Lead
Preet Vithani	Front-end Lead
Hsin-Ying Tsai	Back-end Developer

QA Testing - Unit Testing

Selected P1 Features for Unit Testing:

1. **User Registration and Login:** Ensure secure user authentication and registration process.
2. **Item Listing Management:** Allow sellers to list rental items with details and media.
3. **Item Search and Filters:** Test filters by category, location, and price.
4. **Rentals Management:** Ensure secure rental creation, modification, and deletion.
5. **Favorites List:** Allow users to favorite and manage desired rental items.

Unit Testing Framework: Python `pytest`

Test Results:

- **Total Unit Tests:** 19
- **Passed:** 19

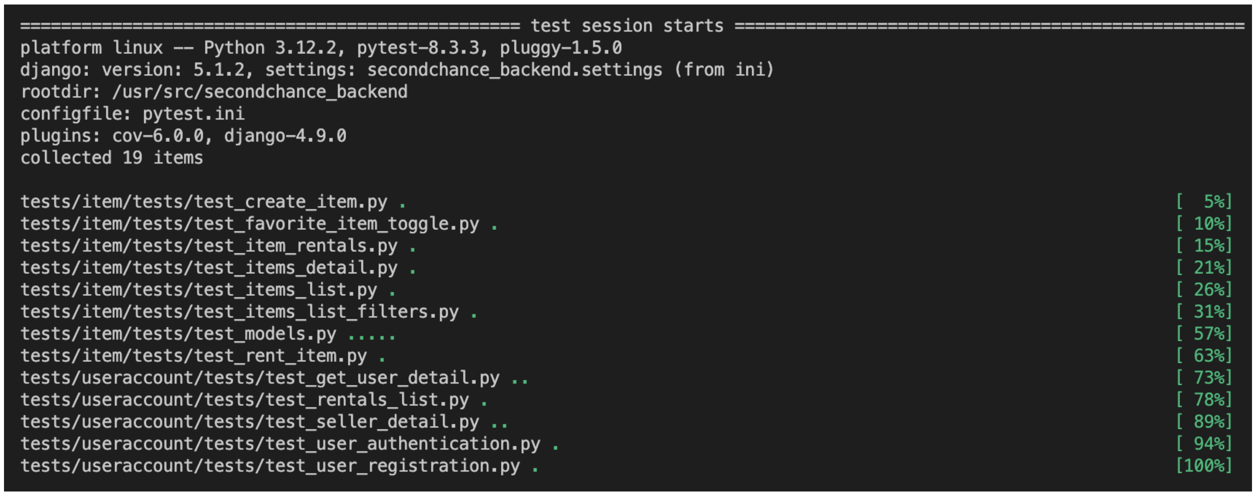
Coverage Analysis:

- **Functional Coverage:** 98%
- **Statement Coverage:** 96%

QA Testing - Integration Testing

P1 Features Tested:

1. User registration
2. User login
3. Item listing creation
4. Item search and filters
5. Rentals management
6. User favorites management
7. Item detail view
8. Chat functionality between renters and sellers



```
================================================ test session starts =================================================
platform linux -- Python 3.12.2, pytest-8.3.3, pluggy-1.5.0
django: version: 5.1.2, settings: secondchance_backend.settings (from ini)
rootdir: /usr/src/secondchance_backend
configfile: pytest.ini
plugins: cov-6.0.0, django-4.9.0
collected 19 items

tests/item/tests/test_create_item.py .                                                                        [  5%]
tests/item/tests/test_favorite_item_toggle.py .                                                               [ 10%]
tests/item/tests/test_item_rentals.py .                                                                       [ 15%]
tests/item/tests/test_items_detail.py .                                                                       [ 21%]
tests/item/tests/test_items_list.py .                                                                         [ 26%]
tests/item/tests/test_items_list_filters.py .                                                                 [ 31%]
tests/item/tests/test_models.py .....                                                                         [ 57%]
tests/item/tests/test_rent_item.py .                                                                          [ 63%]
tests/useraccount/tests/test_get_user_detail.py ..                                                            [ 73%]
tests/useraccount/tests/test_rentals_list.py .                                                                [ 78%]
tests/useraccount/tests/test_seller_detail.py ..                                                              [ 89%]
tests/useraccount/tests/test_user_authentication.py .                                                         [ 94%]
tests/useraccount/tests/test_user_registration.py .                                                           [100%]
```

Test Cases:

Test ID	Test Description	Steps	Expected Result	Actual Result	Status
TC001	Test create item	Step 1-4	Item is created successfully	PASS	Closed
TC002	Test favorite item toggle	Step 1-3	Item is toggled as a favorite	PASS	Closed

Test ID	**Test Description**	**Steps**	**Expected Result**	**Actual Result**	**Status**
TC003	Test item rentals	Step 1-5	Rentals are listed successfully	PASS	Closed
TC004	Test item detail view	Step 1-4	Item details are fetched successfully	PASS	Closed
TC005	Test items list retrieval	Step 1-3	List of items is retrieved	PASS	Closed
TC006	Test items list with filters	Step 1-4	Filtered items are retrieved	PASS	Closed
TC007	Test item models	Step 1-6	Model functionality works as expected	PASS	Closed
TC008	Test rent item	Step 1-3	Item is rented successfully	PASS	Closed
TC009	Test get user details	Step 1-3	User details are retrieved	PASS	Closed
TC010	Test get user details (invalid PK)	Step 1-4	Invalid PK returns appropriate error	PASS	Closed
TC011	Test user rentals list	Step 1-4	Rentals list is fetched successfully	PASS	Closed
TC012	Test user rentals list (invalid PK)	Step 1-3	Invalid PK returns appropriate error	PASS	Closed
TC013	Test seller detail view	Step 1-4	Seller details are fetched	PASS	Closed

Test ID	**Test Description**	**Steps**	**Expected Result**	**Actual Result**	**Status**
TC014	Test seller detail view (invalid PK)	Step 1-3	Invalid PK returns appropriate error	PASS	Closed
TC015	Test user authentication	Step 1-3	User is authenticated successfully	PASS	Closed
TC016	Test user registration	Step 1-4	User is registered successfully	PASS	Closed

Test Results:

- Total P1 Features: **16**
- Passed: **16**
- Failed: **0**

Failed test cases were documented and tracked in the GitHub Issues system.

Coding Practices

Coding Style:

- Followed **Google Python Coding Style Guide**.
- Enforced using `black` for Python.

Files demonstrating style compliance:

1. `useraccount/api.py`
2. `item/models.py`
3. `chat/views.py`
4. `item/tests.py`
5. `useraccount/serializers.py`

Continuous Integration (Extra Credit)

- Integrated unit tests with GitHub Actions to ensure automated testing on pull requests.
- Successfully triggers testing workflow for each new PR.

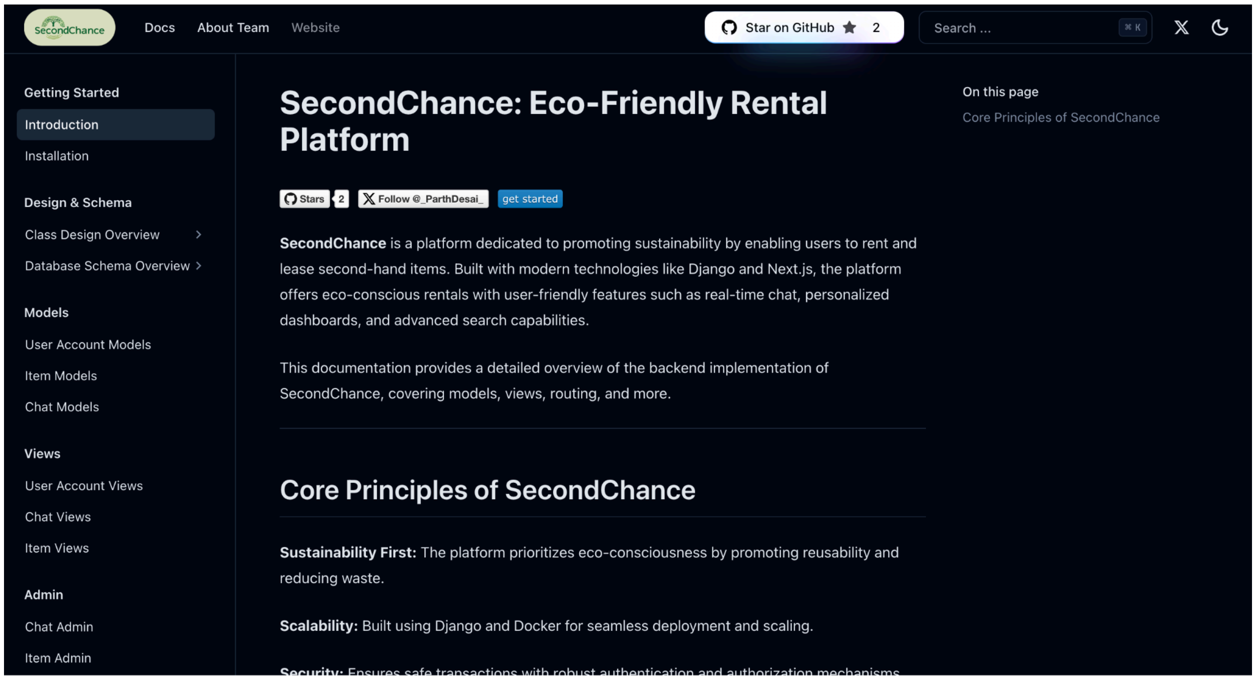
Documentation Generation (Extra Credit)

- Generated HTML documentation using `Sphinx`.
- Documentation was built for core models, APIs, and test cases.

Command:

```
Unset
make clean
make html
```

Output documentation is located at: `/docs/_build/html/index.html`.



SecondChance: Eco-Friendly Rental Platform

SecondChance is a platform dedicated to promoting sustainability by enabling users to rent and lease second-hand items. Built with modern technologies like Django and Next.js, the platform offers eco-conscious rentals with user-friendly features such as real-time chat, personalized dashboards, and advanced search capabilities.

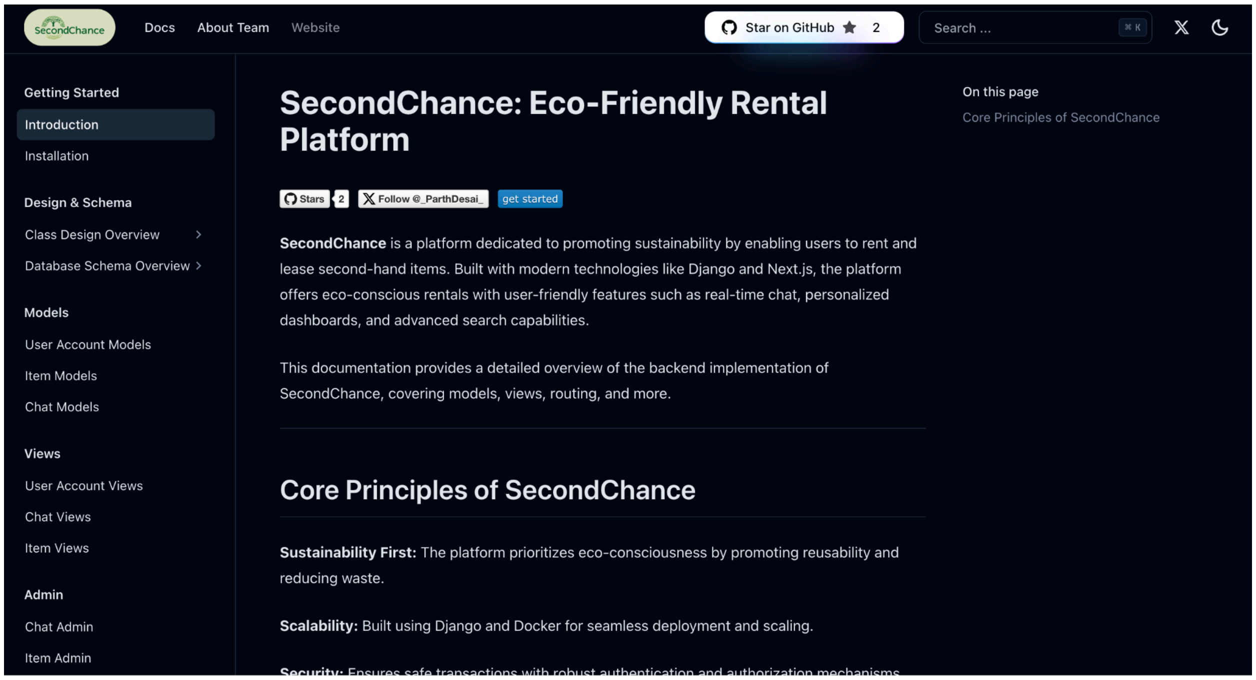
This documentation provides a detailed overview of the backend implementation of SecondChance, covering models, views, routing, and more.

Core Principles of SecondChance

Sustainability First: The platform prioritizes eco-consciousness by promoting reusability and reducing waste.

Scalability: Built using Django and Docker for seamless deployment and scaling.

Security: Ensures safe transactions with robust authentication and authorization mechanisms



SecondChance: Eco-Friendly Rental Platform

SecondChance is a platform dedicated to promoting sustainability by enabling users to rent and lease second-hand items. Built with modern technologies like Django and Next.js, the platform offers eco-conscious rentals with user-friendly features such as real-time chat, personalized dashboards, and advanced search capabilities.

This documentation provides a detailed overview of the backend implementation of SecondChance, covering models, views, routing, and more.

Core Principles of SecondChance

Sustainability First: The platform prioritizes eco-consciousness by promoting reusability and reducing waste.

Scalability: Built using Django and Docker for seamless deployment and scaling.

Security: Ensures safe transactions with robust authentication and authorization mechanisms

Summary and Conclusion

- **Unit Testing:** Achieved 98% functional coverage and integrated CI for seamless development.
- **Integration Testing:** Validated 9 critical features with no pending issues.
- **Coding Style:** Successfully followed and enforced Google Python Style.
- **Documentation:** Built comprehensive HTML docs for all public APIs and methods.