Implementation of LatticeQED problem:

1. most general case: quantum link model (with gauge fields) in d dimensions (limit to 1 or 2 dimensions first)
2. as special cases:
   - Wilson and staggered fermions
   - in 1D: possibility to eliminate gauge fields
3. start with Wilson Hamiltonian in 1D and 2D + QLM for gauge fields
4. need to enforce Gauss' law: e.g., as an additional Hamiltonian term
5. User should have the following options to specify:
   - lattice (includes: dim, lattice_constant, num_sites)
   - fermion_rep (Wilson or staggered)
   - physical_params (mass, coupling, etc.)
   - whether to add Gauss' operator to Hamiltonian (energy penalty) or return as a separate operator
   - for fermion_rep == 'wilson': representation of gamma-matrices
   - for dim == 1: whether to eliminate gauge fields
6. combinations of operators:
   - FermionOp * FermionOp
   - FermioOp * SpinOp * FermionOp
   - SpinOp * SpinOp
7. introduce new operator.second_quantization.mixed_op
   - mixing/composition is always as a tensor product since different types of operators must act on distinct Hilber spaces
   - MixedOp = List[FermionicOp, SpinOp, FermionicOp]
   - need to specify WHICH qubits each op acts on? Allow for cases where two FermionicOps act on different registers?
   - can this be made compatible with the current implementation of mappers?
8. return list of operators as
   - List[MixedOp(FermionOp, FermionOp), MixedOp(FermionOp, SpinOp, FermionOp), ...], each element is term in operator